

Houses In Reach
A Personal Real Estate Monitoring and Mining Application

by

Sweta Gupta

MCA, Barkatulla University, 1997
B.Sc., Awadesh Pratap Singh University, 1994

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

© Sweta Gupta, 2008
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.

Houses In Reach

A Personal Real Estate Monitoring and Mining Application

by

Sweta Gupta

Supervisory Committee

Dr. Hausi A. Müller, Supervisor (Department of Computer Science)

Dr. Ulrike Stege, Departmental Member (Department of Computer Science)

Dr. Jens H. Weber, Departmental Member (Department of Computer Science)

Dr. Jochen Moehr, External Examiner (School of Health Information Science)

Supervisor: Dr. Hausi A. Müller

Abstract

The information technology, which is inescapably penetrating all facets of industry and our lives, is propelling the real estate industry into unknown territory. The number of websites providing access to information about property and its environs is steadily increasing. By automating various processes, cost-effective websites offer high quality services. Despite on-line access, one has to analyze a seemingly infinite variety of information to find desired houses or properties. Most buyers, who use the web to investigate real estate, login and re-login to many websites to keep track of houses, which is rather time consuming. Sometimes properties of interest have sold even though a buyer is still waiting for the price to come down. Moreover, real estate websites typically provide large lists of houses without taking the preferences of a particular customer into account. Some websites use proprietary protocols and formats to store and publish house listing data. While these formats are easy to read, there is no common format and thus making it difficult for web developers to consolidate data from different sources.

The goal of our research is to design a flexible and extensible architecture for a real estate engine to be able to draw data from different real estate sources effectively. In addition, it is important to design this engine to display properties based on user preferences, rather than merely providing a list of all properties currently available.

In this thesis, we present the design and implementation of the components of our “Houses In Reach” real estate engine, which addresses the problems mentioned above. The main components of this web engine include an architectural model, search engines to look for pertinent information on the web based on user preferences, and a visualization engine to display houses on Google maps. The thesis concludes with a discussion of our experience building these components.

Examiners:

Dr. Hausi A. Müller, Supervisor (Department of Computer Science)

Dr. Ulrike Stege, Departmental Member (Department of Computer Science)

Dr. Jens H. Weber, Departmental Member (Department of Computer Science)

Dr. Jochen Moehr, External Examiner (School of Health Information Science)

Table of Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	v
List of Tables	vii
List of Figures	viii
Acknowledgments	ix
Dedication	xi
Chapter 1	1
Introduction	1
1.1 Motivation.....	1
1.2 Approach.....	6
1.3 Contributions.....	8
1.4 Thesis Outline.....	10
Chapter 2	11
Related Work	11
2.1 Features required for an effective real estate website.....	11
2.2 Features of current real estate websites.....	12
2.3 Technologies available for real estate websites.....	14
2.4 Features and tools not available for real estate websites.....	15
2.5 Improvements and suggestions.....	16
2.6 Summary.....	17
Chapter 3	19
Literature Review	19
3.1 A model for real estate websites.....	19
3.2 User preferences model.....	20
3.3 RSS Feeds and RSS Reader for real estate website applications.....	24
3.4 Summary.....	29
Chapter 4	30
System Architecture	30
4.1 A model for real estate website applications.....	30
4.2 Detailed component description for a real estate website application.....	32
4.3 Summary.....	36
Chapter 5	37
The Visualization Engine	37
5.1 Fundamentals of Google Map API.....	37
5.2 Use of Google Map API for a website.....	37
5.3 Comparison with the Yahoo Map API.....	43
5.4 Limitations and difficulties.....	44
5.5 Summary.....	44
Chapter 6	46
Monitoring Utility	46
6.1 Importance of monitoring for a real estate website.....	46
6.2 Monitoring for Houses In Reach website.....	47
6.3 The Limitations of Houses In Reach monitoring service.....	52

6.4 Summary	53
Chapter 7	54
Parsing Engine	54
7.1 Registration of RSS feed.....	54
7.2 Detail of Parsing Engine	60
7.3 Difficulties and Limitations	68
7.4 Summary	68
Chapter 8	70
User Preferences.....	70
8.1 Data model for real estate website for user preferences	70
8.2 User preferences algorithm	73
8.3 Use of user preferences for Houses In Reach website.....	75
8.4 Summary	80
Chapter 9	81
Evaluation.....	81
9.1 Cognitive Dimensions.....	81
9.2 Evaluation of Parsing Engine.....	85
9.3 Evaluating the Visualization Engine.....	88
9.4 Evaluating the Monitoring Utility.....	91
9.5 Summary	93
Chapter 10	94
Conclusions.....	94
10.1 Summary	94
10.2 Contributions.....	95
10.3 Future Work	96
Bibliography	98
Appendix A Glossary	102
Appendix B Set Up Instructions	104
Appendix C System Design	106
Appendix D Home Buyer Scenario	120
Appendix E Selected Houses In Reach Code.....	126
Appendix F User Preference algorithm	131

List of Tables

Table 1. Major Web Characteristics for Real Estate.....	5
Table 2. The Ten Most Visited Websites and Their Features.....	13
Table 3. Life Values and Descriptions [LiGM1989].....	21
Table 4. Housing Attributes [LiGM1989].....	22
Table 5. Perceived Effects of Housing Attributes on Life Values [LiGM1989].....	23
Table 6. Sample RSS Feed stream.....	26
Table 7. Comparison of Google and Yahoo Map APIs.....	43
Table 8. The XML for the Feed of Figure 17.....	62
Table 9. The Value Parsed by ROME.....	64
Table 10. Database Field Name and Value.....	64
Table 11. The Value of Description Tag.....	66
Table 12. User Preferences Data Model [LiGM1989].....	71
Table 13. Basic Cognitive Dimensions [GrB11998].....	82
Table 14. Questions Regarding Cognitive Dimensions [B1Gr2000].....	83
Table 15. Questions Regarding Cognitive Dimensions [B1Gr2000] (Continued).....	84
Table 16. Comparison of Parser Form and Zillow Feed Registration Form.....	86
Table 17. Comparison of Visualization Engine and Zillow's Mapping Service.....	90
Table 18. Comparison of Monitoring Service with Zillow Monitoring Service.....	92

List of Figures

Figure 1. Conventional Real Estate Transactions	2
Figure 2. Property Information Provided When Listing a Website	4
Figure 3. The Conventional Methods of Real Estate Trading	6
Figure 4. Basic Components for a Real Estate Website Application	8
Figure 5. Email Update by Zillow	15
Figure 6. Four-Layer Web-Based Real Estate Model by IMW [AnAA2004].....	19
Figure 7. Interaction between Websites, Web server, RSS Reader, and RSS feed	25
Figure 8. Architecture of ROME [Java2007]	28
Figure 9. Four-Level Web-Based Real Estate Model.....	31
Figure 10. Sample Google Map with Information Window	41
Figure 11. Record of Price History and Price Graph	48
Figure 12. New House Entry in User Account	49
Figure 13. The Change in Price Notation	50
Figure 14. The “Deleted” Notation.....	51
Figure 15. The Results Page	52
Figure 16. RSS Registration Form.....	56
Figure 17. Direct RSS Feed Parsing Approach	59
Figure 18. RSS Feed Parsing Approach Using Level of Indirection.....	60
Figure 19. A Sample RSS Feed	62
Figure 20. Intrinsic Attributes.....	76
Figure 21. Location Attributes	77
Figure 22. Life Values	78
Figure 23. Houses for Life Value “Comfort”	79
Figure 24. Houses for Life Value “Togetherness”.....	79
Figure 25. Houses for Life Value “Family”.....	80
Figure 26. Zillow Feed Registration Form	85
Figure 27. Zillow’s Visualization Engine.....	89
Figure 28. Zillow Monitoring Service	91
Figure 29. Package Diagram for Houses In Reach Prototype	106
Figure 30. Class Diagram for Monitoring Services	107
Figure 31. Class Diagram for User Interface	108
Figure 32. Class Diagram for Generating Chart	109
Figure 33. Class Diagram for setting User Preferences	110
Figure 34. Class Diagram for Parsing RSS feed.....	111
Figure 35. Class Diagram for Checking updates	112
Figure 36. Sequence Diagram for Checking Updates.....	113
Figure 37. Sequence Diagram for Comparing Price.....	114
Figure 38. Sequence Diagram for Checking New Price	115
Figure 39. Sequence Diagram for Checking Deleted Entries	116
Figure 40. Sequence Diagram for Parsing RSS feed	117
Figure 41. Google Map API.....	118
Figure 42. Database Design	119

Acknowledgments

Standing on the brink of completing this thesis, I can't help reflecting on the people who contributed to the path that brought me to this point, and more deeply, to who I am.

I wish to acknowledge Behrokh Goodarzi and Janet Lawlor, who came in my life as two angels and stayed with me from morning until evening through emails, encouraging me, and guiding me during my tough times, cheering me and celebrating the achievement of every milestone.

I do not want to forget Maria-Elena Hernandez's help in clarifying my thoughts about research. She was with me from the beginning of my research constantly encouraging me to continue.

I want to thank Neelam for helping me in designing the webpage and Suhani, for letting her mom work with me

I am grateful to Purvee Yagnik for solving my SQL problems and teaching me the lessons of keeping back ups and helping me to retrieve the database whenever I messed up with the database the nights before my presentations.

I appreciate Qin Zhu for asking hard questions and for being a valuable critic of my research.

I will never forget Ron Desmarais for coming up with the idea of RSS.

Steve Slavik gave me generous help in my writing. Likewise, Arvind Sund, a realtor, provided resources on real estate. Scott Brousseau, Dylan Dawson, Lei Lin, Atousa Pahlevan, Tony Lin, Alexey Rudkovskiy, Sangeeta Neti and Feng Zou made valuable contributions in scoping an interesting, but manageable problem. I thank the rest of my friends Shashi Monie, Raj Monie, Mercy Jackson, Manjeet Johal, Ganseh, Brij Paul, Anil

Singh, Luis Izquierdo, Rukmanee, Pavan Gill and Stephane, who kept teaching me what being human is about.

Of course without my parents I would have never done this. My father constantly encouraged my education and supported my interests. My mother strengthened my resolve to continue my education. My brother Samir Gupta and sister Smita Goyal made me realize how important I am for them and how proud they are of me being their younger sister. I appreciate my beloved nephews Pulkit and Geetansh for asking me every time, “When are you finishing your studies and coming back to India to play with us?”

My late uncle Jawala Gupta, my aunt Meena Gupta and my cousin Anita Singhal trusted me and encouraged me during my critical times and kept up my morale, reminding me that I have a family and they are always with me in whatever circumstances I might be.

Udaya Bhaskar gave me a good start in writing and taught me the techniques of writing. He also helped me in providing resources on real estate literature.

I am deeply appreciative of the faculty in the Department of Computer Science at the University of Victoria for building a truly supportive environment for research and learning.

Finally, and most importantly, I thank Dr. Hausi Müller for helping me in my research. He is a true teacher in the sense that he gave me tools that shaped the way I look at problems and, no less important in engineering, solutions. He also helped me in my research and in providing financial assistance without which I could not have continued this program. I also want to acknowledge his generous nature in dealing with students.

Dedication

To My parents

Sri. Vijay Pal Gupta

Smt. RajKumari Gupta

&

To My Uncle

Late Sri. Jawala Prasad Gupta

(1930 to 2005)

Thank you for your encouragement and your trust in me

Chapter 1

Introduction

People can create food and clothing, but cannot grow another square inch of land. Since the beginning of civilization, it has been the constant endeavour of every person to have a house. After agriculture, housing is one of the milestones in innovation; however, it has become so ubiquitous that it has been taken for granted always to exist. With the invention of the joint stock company, the Dutch and the English introduced new ways of holding wealth at the beginning of the 17th century [WiFG2003]. Since then, land and buildings have become another form of wealth. After the industrial revolution, land and buildings started to become objects of economic importance and have been traded like commodities and services.

Real estate boards were set up in many countries to organize the trade in land and buildings in the United States in 1880 and in Canada in 1888. One of the early systems for real estate trading in 1951 was the Photo-Coop system, a pre-cursor to the Multiple Listing Service (MLS). Both systems were built to organize the real estate trade.

1.1 Motivation

Real estate is a robust investment method. As the global population increases, so does the need for more land and buildings. The demand for real estate increases drastically, while the supply remains the same, with no increase or decrease although reclamation sometimes adds an inch of land. This trade is also not vulnerable to the corrupt practice of investment houses. Real estate has never been valued more than now and never before in history could a common person take advantage of a piece of land as an investment.

Hudson-Wilson *et al.* mention six primary reasons for inclusion of real estate in investments [WiFG2003]:

1. to reduce the overall risk of a portfolio by combining asset classes that respond differently to expected and unexpected events;
2. to achieve absolute returns well above the risk-free rate;

3. to hedge against unexpected inflation or deflation;
4. to constitute a part of a portfolio that is a reasonable reflection of the overall investment universe (an indexed or market-neutral portfolio);
5. to deliver strong cash flow to the portfolio, and
6. to obtain tax benefits.

Wigand and Crowston describe how real estate transactions take place. Transaction processes proceed in five stages [CrWi1999].

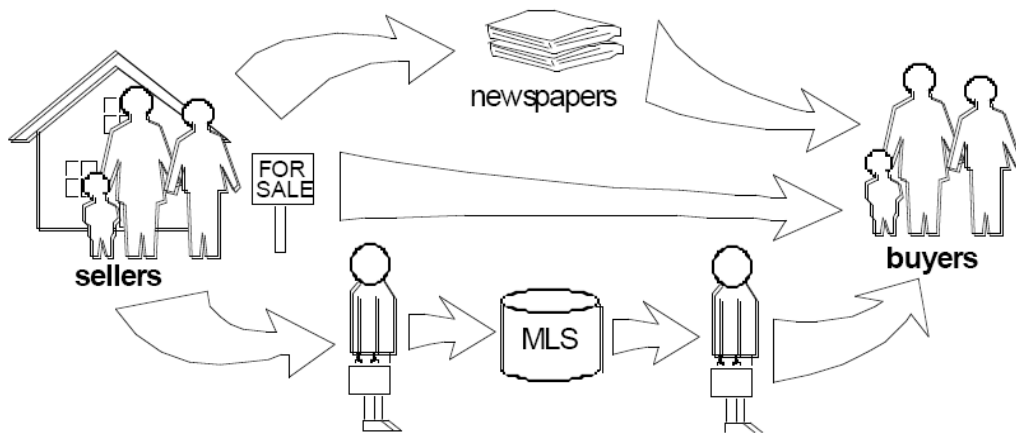


Figure 1. Conventional Real Estate Transactions

- 1. Listing:** Potential sellers put their house on the market. The conventional method includes the use of real estate agents for listing the house in the multiple listing services in exchange for a commission on the sale or advertising a house in the classified section of the newspaper or a sign. This is depicted on the left side of Figure 1.
- 2. Searching:** Potential buyers review houses according to their specific needs. These needs might include specific location, price, and size. Buyers also go to the same media to find the houses as the sellers use to advertise. The process of finding and perusing a house takes a long time. Sometimes, it may not even be possible to visit the house due to distance.
- 3. Evaluation:** Once buyers find a house, they would like to evaluate it in several different ways. This may include an agent's inspecting the house with the buyer.
- 4. Negotiation:** The buyer and the seller negotiate over the price and other conditions.

The agent usually advises the buyer. Offers and counter-offers are traded.

5. Execution: The sale is closed after the legal deeds are transferred. A third party typically finalizes the transfer, for instance, a lawyer of the title company.

This traditional system has some drawbacks. In this system, the agent plays a pivotal role. The buyer expects the agent to research and present houses that are suitable, negotiate the price, complete paper work, calculate purchasing power, and arrange finances. Traditionally the agents charge a large fee from both the buyer and the seller. Only the real estate agents have the power to subscribe to multiple listing services and only the agents can obtain the detailed home listing information necessary to conduct a thorough home search. Since the agents control this listing information, they enjoy a monopoly in terms of information ownership, guaranteeing a significant role in a buyer-seller relationship [CrWi1999]. According to Bond *et al.*, access to listing data represents an entry barrier for the real estate market [BSSB2000]. They argue further that, were this barrier lowered, market power would be taken away from the agents and distributed to sellers and buyers.

With the event of the information super-highway, when access to information is exploding and when the World Wide Web is changing the rules of all trades, real estate is not immune to technology. Information is presented in more than one form on the Web including text, images, sounds, video, and animation. When so much money is involved in a real estate transaction, the Web can make it possible to know more and be more prudent in selling and buying real estate. The conventional ways of the real estate trade make it painful and sometimes a nightmare for both the buyer and the seller. The large waiting times after listing, waiting for a better offer, the ordeals of settlement, and the problems that arise with the sale are some of the difficulties. Some of these can be solved by using the Web for real estate trade. The Web also provides a great convenience as the buyer and seller can sit in the comfort of their living rooms. Through the Web, a buyer may see the inside of the house through virtual tours, talk to the vendor, and find more information about the building, locality, and local amenities, enabling more prudent decisions.

The real estate trade has adopted the technology and the information that was previously controlled by real estate agents and is now freely available. Many websites, for example Realtor.com, sponsored by the Canadian National Association of Realtors (NAR), has approximately 1.2 millions homes listed in various multiple listing sites [Pitt1999].

Bond *et al.* performed a study on how the real estate brokerage industry responds to opportunities of listing the properties on the Web [BSSB2000]. The study showed that an average site has been operational for one and a half years and has attracted as many as 440 hits per day. The outcome of the survey is depicted in Figure 2. The questionnaire for the survey sought such information as geographic information, asking price, square footage, the presence of fireplaces and swimming pools and financial information like property tax, insurance, and association fee. Also the questionnaire sought information on recent comparable sales; information on surrounding schools, shopping and recreation; information of the kinds of images of the houses available; and other pertinent information like mortgage calculator and pre-qualification.

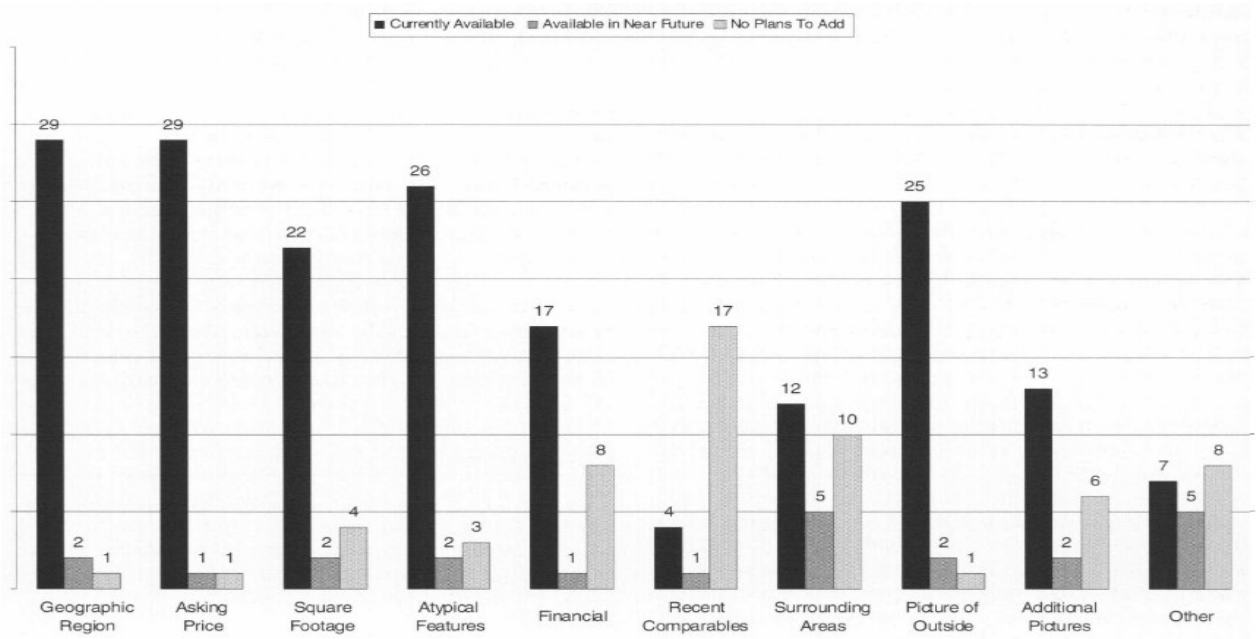


Figure 2. Property Information Provided When Listing a Website

The effects of the Web on real estate firms were studied by Bardhan *et al.* [BaJK2000]. Table 1 summarizes their findings. The advantages of the Web influence real estate businesses in their property management or development, relocation services, real estate finance services, and listing or matching services, and other information management.

Table 1. Major Web Characteristics for Real Estate

Internet /Web Characteristic	Advantages / Opportunities
Interactive Search Capability: Graphics Capability, Virtual Tours, Geographic Outreach	Improves communications within firms and with customers or clients. Expands geographic range within which a firm can operate.
Creation of online communities and markets	Increases market efficiency and depth, Business to Business coordination for new kinds of transactions.
Online Tracking and Analysis	Product and service information can be provided at a depth and complexity that is not possible in other media.
Web Based Interactive Transactions	Provides speed, low marginal costs, and interactive component. Firm may be able to change staffing needs or the geographic location of staff.

Through the use of the Web, real estate marketing changes to something like that depicted in Figure 3 with the grey arrows, where For-Sale-by-Owner (FSBO) adds to the other sources of real estate trade resources and advertising. The agent is still present and still seeks a commission for bringing a buyer to the seller, but the listing is advertised at multiple places, helping to bypass the agent.

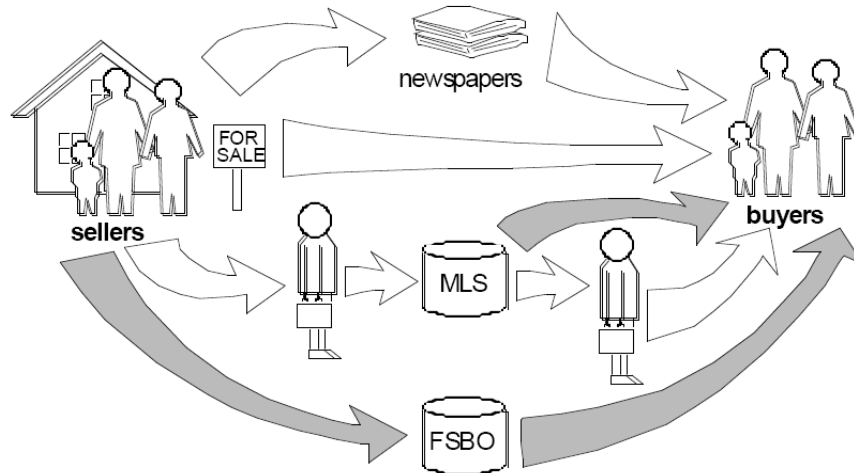


Figure 3. The Conventional Methods of Real Estate Trading

Despite the developments in technology that can make real estate trade a pleasure, or at least not so painful, there is room for improvement. For instance, despite on-line access, one has to analyze all the seemingly infinite information to find the house or property that one wants. Sometimes, the property might have been sold, or the buyer might not know in advance the different issues with it. The time has come for an information sieve that can inform the user of the choices that fit his or her needs or specifications.

1.2 Approach

Trading in real estate is very time-intensive. Because considerable money is involved, the buyer and the seller want to ensure that they are making the best possible choice. The buyer wants to buy a house that fits his or her needs and dreams as much as possible. When they do not find an exact match, they will usually settle for something near. By providing information pertinent to the needs of the buyer and allowing more information to flow through the system, it is possible to make the buyer and the seller happier in the end. It is preferable for the buyer to spend less time in the different stages of buying houses and to let him or her converge on the right buying choice by providing information on only those houses that fit his or her needs. Further, it is better if the buyer does not have to log into every listing website and check for new listings, go through them, find those interesting, and proceed further. Instead, he or she may be directed to a

website that can be maintained in such a way that he or she has only to log into this site and find all the listings that match his or her preferences. Emailing the buyer about potential candidates when they arrive is also a good practice since nearly 80% of Web users check their mail-boxes every hour [InWS2007].

Buyers may also be provided with updates about their chosen houses. Sometimes, after considerable effort, they find that a property was sold or that the seller did not want to sell it anymore. Hence, updates can save users time. In addition, often listings are duplicated as the seller lists with many agents in the hope of sealing a good deal as soon as possible. Therefore, the information also needs consolidating.

The consolidated information can be displayed on the map. Presenting real estate information on map can help buyers examine their choices that match their requirements compared to the conventional ways of inspecting in person each potential property. Thus, it will save time and money and improve the process of buying a house.

In addition, the increase in the number of websites has also increased competition among realtors. To survive in this competitive environment, realtors have to assist users efficiently, at lower prices. By automating various house hunting procedures within the enterprise, cost-effective websites can offer high quality services.

Thus, the objective of this thesis is to develop a personal real estate engine that promotes, facilitates, and enhances the buying and selling of real estate by providing access to and use of real estate listings more personal, user-friendly, timely, and effective [Seth2003]. The specification of such a personal real estate engine might include a real estate website model, a component to search for pertinent information on the Web and respond with a listing of houses based on user preferences. Once the selection of houses is made, the component monitors the chosen houses. This monitoring service is responsible for any changes made in the house listing. The changes could be new listings of houses, sold houses, or changed prices for the houses selected by the user. This monitoring service would notify the potential buyer of any new houses that match her or

his requirements or preferences.

1.3 Contributions

The real estate engine designed and implemented for this thesis is named “Houses In Reach” and consists of four major components that perform different roles, as shown in Figure 4.

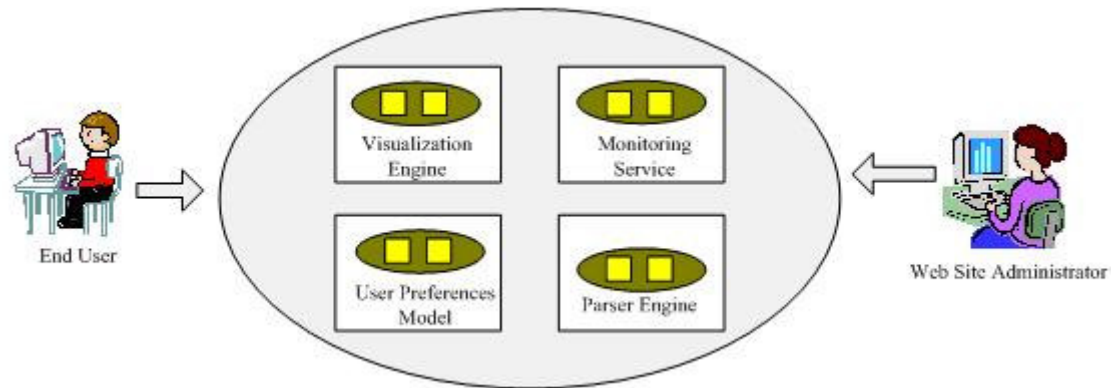


Figure 4. Basic Components for a Real Estate Website Application

1. Parsing Engine: This component reads and filters the data from Really Simple Syndication (RSS) feeds that come from different listing sites and place the listing in a local database. Thus, multiple RSS feeds are supported through a single website. These feeds are available in different text formats and are registered through a form. This form is designed to support a level of indirection by referring the different text format RSS feeds items to the Java String functions and saving it to the a database table. It helps to separate the data source which is the RSS feed with the programs operating on the data which is the monitoring service. Thus the data source and the data integrator are independent to each other. The parsing engine helps to create that separation through this registration form and monitoring service is the program that runs on the integrator.

2. Monitoring services: This component is used to implement a monitoring service to find new listings, keep track of price trends, and send alerts if a house for sale is no longer available in the real estate market.

3. User Preferences Data Model: Based on a study by Lindberg *et al.*, the buyers' preferences are considered and the listings are evaluated based on attributes that are prominent and important [LiGM1989]. For example, a homebuyer making the purchase decision alone may use a different set of criteria than a buyer who is making the decision as part of a family. The proposed model will evaluate properties differently depending on whether a homebuyer is purchasing a house as an investment for his or her parents.

4. Visualization Engine: This component lets the user select and save houses for monitoring. The houses are displayed according to the preferences set by the user. The interface for selecting houses uses the Google Maps interface to locate places visually where houses are for sale. This interface also provides brief housing information and links to more detailed information.

The main contributions of this thesis include:

- Design and implementation of the Houses In Reach real estate engine, a prototype demonstrating that the goal of designing a flexible and extensible architecture for a personal real estate engine to draw data from different real estate sources can be achieved effectively.
 - real estate model to accommodate different RSS real estate sources;
 - user preferences model based on Lindberg *et al.* [LiGM1989] to display houses matching user preferences.
 - customization model; and
 - visualization model using Google Map;
- The implementation of a parsing engine to consolidate RSS feed information and provide a test bed of real time data for real estate engine web developers.
- An innovative architectural solution on how to add additional RSS feeds by end-users using a level of indirection.
- Design of a strategy for integrating different models to achieve data, control and presentation integration for a real estate mash up.
- Design of a strategy for integrating different technologies to implement the Houses In Reach prototype based on the Internet Media Works (IMW) model.

- Design of a personal monitoring utility based on autonomic computing ideas (preliminary).
- Design of rudimentary support for identifying trends with respect to houses and areas (preliminary).

1.4 Thesis Outline

Chapter 2 discusses related work and Chapter 3 provides background information. Chapter 4 discusses a system design that includes a real estate data repository for RSS and discusses RSS feeds as a tool. Chapter 5 discusses the description of the visualization engine, its uses, and difficulties. Chapter 6 discusses the monitoring utility implemented for the real estate website. Chapter 7 elaborates on the parsing engine, describing the retrieval of data from the RSS feeds from different sources by means of a level of indirection, and saving it to the local database. Chapter 8 introduces a user preferences model for listing houses based on a buyer's criteria. Chapter 9 evaluates the major components of the real estate engine using an existing cognitive framework. Finally, Chapter 10 presents conclusions, highlights contributions, and discusses future work.

Chapter 2

Related Work

This chapter provides background knowledge relevant to this thesis and summarizes related work that inspired the research. We first present features that would make an effective real estate website. Subsequently, we describe the resources, features and tools available on current real estate websites. Third, we discuss important features that are not available on real estate websites. Finally, we suggest improvements to incorporate missing features.

2.1 Features required for an effective real estate website

Over the last three decades in general and in the last one and a half decades in particular, the advent of the electronic market and advertising has changed the conduct of commerce. Almost anything may be sold and bought online. With different media of expression and appeal, including graphics, video, and audio, available at the demand of the user, the contents of websites that present commercial information must be structured and easy to navigate and use.

A clear structure and ease of use are as important for websites presenting real estate information as for those presenting any other commodity or service. Gunter has identified three key features that a commercial website must have [Gunt2007]:

1. A clear message: Most users do not go beyond the home page of most websites when the website is not user-friendly. Hence, the home page, the front gate of the website, must be visually and textually well structured and appealing.
2. A client capturing device: At least one device or tool must address the problems that users meet in buying goods or services on that website. These tools may include an e-book of the catalogue, or an audio recording (such as a tutorial). These tools may be more comprehensive than a single web page or a portal.
3. A stay-in-touch mechanism: The client should be followed up for further advice and action. The client must be able to interact with the portal, perhaps in a non-intrusive way. A forum or a blog posting are tools of this kind.

Most websites for real estate have these three basic features, in one form or another.

2.2 Features of current real estate websites

In addition to the three basic features discussed above, many websites have additional facilities that help users narrow their searches by entering a preferred price range, style of property, age of the home, and other features of the property [HeCo2002]. They may also allow searching by proximity to markets and other amenities. For example, www.cyberhomes.com, www.zillow.com, and www.homegain.com are websites that allow searching in these fields. As well, websites listing houses by for-sale-by-owner (e.g., www.owners.com, www.by-owner-ol.com, www.fsbo.net, and www.fisbos.com) often have more functions than the basic three listed above. As an aside, some of these websites are funded through advertisements and offer free listings for users.

Table 2 presents the ten most visited real estate websites and their features in the US and Canada [HeTu1979]. The columns represent the features for the ten websites available. Besides narrowing down a search, other features are common; *Neighborhood Information* provides a facility to examine prospective communities. *House Value* provides a feature for estimating home values for listed houses and recent home sales for comparable and nearby homes. The *Mortgage Calculator* calculates on-line discount mortgages. The *Update Notification* provides automatic notification to buyers, if new houses matching their preferences come onto the market. The *API* or *Tools* provides tools for real estate professionals, webmasters, real estate website operators, and blogs. It also allows users to access local real estate data such as sales prices, school information, and popular neighborhoods. *Listing Display Based on Preferences* displays houses according to user choices. *Free Service* consists of no-charge services on a website.

Table 2. The Ten Most Visited Websites and Their Features

Website Name	Narrow Search	Neighborhood Information	House Value	Mortgage Calculator	Notify/ Update Listing	API Available For Integration	Listing Display Based on Preferences	Free Service
Realtor.com	X	X	X	X				X
MSN Real Estate	X	X	X	X				X
Home Gain	X	X	X	X	X			X
Rent.com	X					X		X
Yahoo! Real Estate	X	X	X	X	X			X
AOL Real Estate	X	X	X	X				X
REMAX Real Estate				X				X
Zillow.com	X	X	X	X	X	X		X
RealtyTrac				X				X
Apartments.com	X							X

Note: “X” indicates that the service is available at that website

2.3 Technologies available for real estate websites

The growing demand of the online real estate business has created a new market for application development. Real estate companies and realtors sometimes purchase an application or third party tool to enhance the power of their websites. Some websites develop proprietary protocols and store listing data in that format to publish listings on their website and sometimes encourage other website developers to use that proprietary format. The most common format is RSS (Really Simple Syndication) Web feed as mentioned by Meyers [Mey2007]. The websites, www.zillowfeeds.com and www.trulia.com, have developed their own real estate RSS Web feed¹ formats called Zillow Interchange Format (ZIF) and Trulia Feed Format (TFF).

Some companies provide tools for comparative analyses of price, mapping, and property listing for real estate websites. These tools typically provide a graph of the prices of properties for any location a visitor may choose, and an interactive web map to show the real-estate listing or listing details of properties associated with any area. The websites, www.zillow.com, www.homegain.com, and www.trulia.com provide such tools.

“The most important thing you want to know in real estate is location, location, location,” said Charlene Li, an analyst with Forrester Research and pretty much any other real estate professional agrees with that statement; So how can we understand location if there are no maps? [Cook2005]. A few companies, for example, Yahoo, Google, Microsoft, Trulia, and Ask, provide free map APIs to embed within real estate websites [Grub2006]. Google offers the most popular API and the service most often used for mashups². For example, Google maps may add location information to real estate data as discussed by Craigslist website [Wiki2007].

¹ A **web feed** is a data format used for providing users with frequently updated content.

² A **mashup** is a web application that combines data from more than one web application.

Some websites like www.zillow.com and www.homegain.com provide updates of the houses a user has selected for future reference through emails. HomeGain and Zillow both have the feature to subscribe for email updates to the user. HomeGain will send new MLS updates for the buyers of the agents and Zillow sends monthly updates as shown in Figure 5.



Figure 5. Email Update by Zillow

2.4 Features and tools not available for real estate websites

Many current real estate websites do not utilize the maps such as www.craigslist.org. Although most websites have forms to ask for an e-mail identification, to maintain a long-term relationship with visitors, it is doubtful that a visitor would willingly part with his or her contact information when unsure of the benefits the website provides [Gunt2007]. For example, www.househunt.com provides listings for almost all regions of the US, but visitors can access the listings only if they provide their e-mail information.

Moreover, many websites do not have listings for complete house information. For instance, the website, www.by-owner-ol.com, directs buyers to properties in the US, but only properties in Florida, Los Angeles, and Georgia are available. Also other websites, have missing information, like www.owners.com, provide sufficient information for a house, but many of its listings do not have photos. The website, Zillow, at one point was considered to have the maximum number of hits, but according to a published report, it could not provide information for all houses listed [Scho2007]. Agents, owners, appraisers, and many others have spotted many errors in Zillow's information. [Bett2007], [Mill2006], [Cook2006], [Matt2007].

Many website users want to have a facility to interact with a website listing, so that they can view the statistics of their listings. For example, www.homegain.com provides a home value, but lacks a feature where the user can save the house listing for future reference. As well, the website provides comparable prices, but the comparable properties are not necessarily on the same street as the desired house. Also, recent house listings may be a year old. Thus, it is very difficult to say that these houses are truly comparable.

In summary, data are often too thin to be reliable and the user has no way to know whether the data is correct or not. Although the price index is important, it is not the most important item on the list of features. Combining both realtor listings and for-sale-by-owner listings means that buyers may not be aware of the status of houses they are considering. HomeGain and Zillow allow house listings for free, but this does not provide any validation process. Anyone could post a false posting. In addition, if a user wanted to buy a house for the sake of health or for his or her family, then such suitable information is difficult to ascertain.

2.5 Improvements and suggestions

We can make a number of suggestions for improvements to the typical real estate website. The improvement that we suggest is for a web application to display properties based on user preferences, rather than a list of all properties available. For example, HomeGain provides a list of all houses even when a user narrows down the search. Some houses lists are not even close to user choices. The user has to select manually the houses from a list provided. Thus, we need a feature to display houses based on user preferences.

Real estate website developers require a tool to test features like the display of property listings in a real time environment. RSS feeds are available free of charge and can provide an environment for testing purposes. Zillow allows users to register the feed, with their own proprietary feeds using ZIF and allows parsing and saving of data at the server side. Most of the real estate website supports RSS feed as compared to

ZIF. Hence we could provide an API that would help register, parse, and save data locally for any real estate RSS feed.

The display of houses on maps requires improvement, for example, the satellite map used by Zillow has errors; it often erroneously identifies a property as someone else's property [Kern2007]. Also Zillow provides a housing display on the map and its brief description at the bottom of the page, leading to user confusion. This could be improved by using Google map APIs. With its help, we could display the brief housing information in the marker window on the map and avoid replicating the information on the same web page.

The websites, www.homegain.com, www.zillow.com, www.oodle.com, and www.trulia.com provide automatic update by e-mail. We could extend this feature by not only sending e-mail notification, but also by providing a visual engine for selecting and adding the recommendations for tracking the update. This would also help increase the traffic to a website.

In addition to improving features and proposing a new API, we have used the real estate model proposed by IMW to implement our web site. CommRex, is another commercial real estate exchange website, has successfully adopted an extended model for managing multi-organization data from distributed websites [AnAA2004].

2.6 Summary

The fact remains that despite all the bells and whistles of a website, its most important part is the information it provides about a listing, timely updates of information, and notification to the user of changes in a listing. Few websites concentrate on these aspects well. The websites, www.homegain.com, www.zillow.com, www.oodle.com, and www.trulia.com, are examples of the few that provide these features. Yet combining MLS listings with those of properties for sale by their owners, displaying non-relevant houses on maps, and overloading e-mail traffic with unnecessary

information only adds more confusion and complication to the house-hunting process. To improve the quality of websites, we need to add new features.

Chapter 3

Literature Review

This chapter reviews the knowledge for our thesis. We first describe the real estate model proposed by Internet Media Works (IMW). Subsequently we describe the user preferences approach proposed by Lindberg and close with brief description of RSS and ROME, a RSS reader.

3.1 A model for real estate websites

Every successful e-commerce story indicates that to be successful, an e-business must have a feasible business plan that is based on an appropriate relevant model. To make an e-business successful for real estate, it is important to know what kind of business model to adopt and how to transform it into a web-based application. The model adopted here is that of Internet Media Works (IMW), as shown in Figure 6 IMW has specialized in web-based application implementation, database integration, and web development. This model is effective in sharing multi-organization data from distributed websites [AnAA2004].

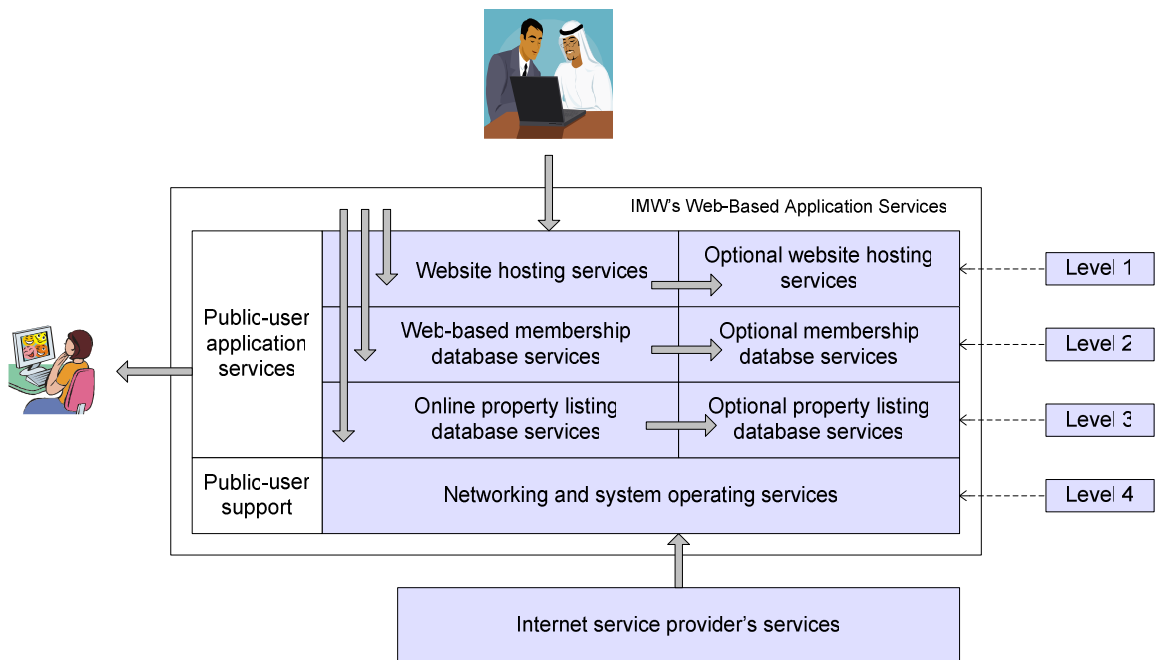


Figure 6. Four-Layer Web-Based Real Estate Model by IMW [AnAA2004]

The model proposed by IMW has a four-level structure. The *first level* is a website structure with HTML, JSP (Java Server pages), XML (Extensible Markup Language), Mapping (a service for generating maps on web pages), page styling, and other features. The *second level* holds the membership database, containing contact information for the subscribers. This level allows subscribers to access features for narrowing down searches and for defining different searches (e.g. houses and locations). This level is an extension of the first level.

The *third level* holds the development and maintenance database for online property listings. Here, sellers can add, delete, and modify property listings and buyers can search property listings. This layer is an extension of the previous two layers. Each buyer owns his or her account and search criteria and results are produced based on these criteria. The *fourth layer* is a networking and system operation service. This layer allows web domain and service set-up at other locations.

By adopting this four layer structure, the IMW architecture supports modularity and reusability. The architecture is versatile and flexible for functions and processes required by subscribers for different business needs. This architecture as IMW claimed supports reusable application modules and also this framework permits easy additions or deletions of components. Thus this framework is best suited to our requirement and will be used to implement our prototype, which we named as “Houses In Reach”.

3.2 User preferences model

In the case of performing online transactions for buying a house, buyers can select only from options prepared by the web designer and will receive a list based on the options he or she has selected. For example, if a user selects two bedrooms and one bathroom, then the houses listed will be of this type. However, if a buyer loves nature, he or she has no options for selecting a house with features that promote love of nature. Thus, we need a component that can catch the occupant’s image and can target families who love nature as prospective buyers.

Traditionally, realtors appear to investigate perceived residential features to see if those features come close to the ideal features of homebuyers. As well, considerable research has been done to investigate consumer attitudes and behaviour with respect to particular products or brands. Upmeyer suggested using desirable end-states. For example, freedom, comfort, and happiness, that a consumer strives to attain in life. Values are an element of the consumer's cognitive structure [LiGM1989]. Table 3 names twelve life values and offers a brief description. We adopted these life values in the construction of our Houses In Reach website because these life values help to define the buyer's image and will help to reason their housing preferences.

Table 3. Life Values and Descriptions [LiGM1989]

Life Values	Definition
Comfort	A comfortable life
Excitement	A life of variety & satisfaction of curiosity
Family	Good relations within the family & provision of a stable home for children
Freedom	Independence and freedom of choice
Happiness	Contentment and joy
Health	Having good health
Inner harmony	Feeling free from inner conflicts
Leisure	Doing something meaningful in leisure time
Money	Having sufficient money & being able to afford things
Pleasure	All kinds of pleasure & enjoyment
Security	Not feeling worried or threatened
Togetherness	A feeling of belongingness & loving someone

Life values play an important role in explaining preferences, choices, and satisfaction. The role of life values was studied by Lindberg, Garling, and Montgomery in relation to residential preferences and simulated residential choices [LiGM1988]. According to them, a person's evaluation of given housing alternatives is determined by a combination of the evaluation of different housing attributes. These housing attributes, according to Clark and Clark [ClC11989] and Hempel and Tucker [HeTu1979], fall under three categories, as shown in Table 4.

Table 4. Housing Attributes [LiGM1989]

Housing Attributes	Sub-Categories
Intrinsic Attributes	Cost, Size, Standard (year build, house Style)
Location Attributes	Distance to downtown Distance to friends and relatives Distance to recreation centre Distance to schools and preschools Distance to work centre
Neighbourhood Attributes	Neighbourhood facilities (e.g., shops) Noise Level (interior and exterior) Reputation (safety, neighbours) Transportation (buses)

The life values listed in Table 3 are assumed to be related to at least one of the housing attributes listed in Table 4 and are the elements of the evaluative process. In further study, Lindberg *et al.* proposed three models based on an investigation with different assumptions concerning how beliefs about the attainment of life values affect consumer behaviour [LiGM1989]. These three models were used for predicting preferences within and choices among hypothetical housing alternatives. The role of consumer value structures was investigated by Lindberg in the context of residential preferences and simulated residential choices by means of a questionnaire. The perceived effects of each housing attribute on the possibility of attaining the twelve life values was computed for each of the three models. Table 5 shows the results of their studies. The ratings are evaluated on the scale of -6 (worst) to +6 (best).

Table 5. Perceived Effects of Housing Attributes on Life Values [LiGM1989]

Housing Attributes	Life Values												
	Comfort	Excitement	Family	Freedom	Happiness	Health	Inner harmony	Leisure	Money	Pleasure	Security	Togetherness	<i>M</i>
<i>Intrinsic</i>													
Cost	1.14	3.28	2.39	2.75	1.58	0.86	1.75	2.22	6.22	1.22	1.75	1.06	2.19
Size	5.50	2.03	6.06	4.52	2.86	2.39	3.69	2.08	-2.63	2.78	1.78	3.19	2.85
Standard	3.83	0.25	2.17	1.22	0.86	0.72	0.67	0.25	-2.22	0.75	1.00	0.97	0.87
<i>Location (distance)</i>													
Downtown	3.36	2.17	1.00	2.44	0.53	-1.00	0.69	1.31	1.92	-0.19	0.42	1.44	1.17
Friends	2.28	2.92	2.86	2.69	1.89	0.94	1.97	2.25	1.53	1.61	2.36	4.75	2.33
Recreation	4.08	4.19	3.03	4.25	1.92	3.11	2.11	5.28	2.00	2.03	1.33	3.06	3.03
Schools	3.42	1.89	3.94	3.92	1.36	1.08	2.19	1.89	2.17	1.11	2.44	2.36	2.31
Work	4.36	2.36	3.33	4.00	1.42	1.42	2.39	2.53	3.47	1.42	1.89	2.25	2.57
<i>Neighbourhood</i>													
Facilities	5.31	2.22	3.69	4.52	1.33	1.08	1.69	2.58	2.44	1.58	2.08	2.22	2.56
Noise	4.19	1.56	4.03	2.39	1.97	4.56	3.75	1.72	0.39	2.42	2.28	2.16	2.62
Reputation	1.92	1.92	3.50	2.67	2.53	1.58	3.31	2.00	0.67	2.00	4.08	3.05	2.11
Transport	5.08	4.64	3.89	5.78	1.67	1.22	2.00	4.56	2.72	1.39	2.53	3.72	3.17
<i>M</i>	3.71	2.45	3.32	3.43	1.66	1.50	2.19	2.39	1.56	1.51	2.00	2.52	2.35

M : represents standard Mean

Table 5 depicts the perceived effects of housing attributes on the attainment of life values. The table shows that size of a house is seen as important for the attainment of comfort; size, in turn, is seen as important for the family value as well. Cost was believed to be important for the attainment of money, whereas size and standard attributes were believed to have negative effects on the attainment of money.

The benefit of the Lindberg *et al.* survey is that it is a quantitative analysis. The results shown in Table 5 can be used to rank the housing attributes based on the life values chosen by a website user. If we categorize these numbers to three different levels and search housing attributes for a level from the database it will help to retrieve the houses based on user's preferences. For example, if a user selects "Comfort" as his life value and we range 5.0 to 6.0 as his first preference then we can retrieve all the houses which correspond to the size, facilities and transport housing attributes from the database.

3.3 RSS Feeds and RSS Reader for real estate website applications

The online property listing data has played a major role in Houses In Reach website implementation. The online property listing data is available to us through RSS feeds. First, we define RSS feed and explain its importance for real estate websites. Second, we define the notion of RSS reader, the architecture of ROME and the RSS reader we used for the Houses In Reach website.

Definition and Description of RSS

RSS refers to a collection of Web feeds, specified in XML and published by a website. The main purpose of an RSS feed is to provide:

- Instant Information: information updated in real-time;
- a Single Source: all information aggregated in a single location;
- Rapid Scanning: feed readers highlight unread headlines ; and
- Categorizing: information can be categorized by theme.

Particularly RSS feeds maintain a consistent data format, change their data frequently, and constitute an elegant way to provide updated contents automatically to the end user. The contents can be provided with the help of an RSS Reader. RSS Reader is a computer program installed either at the client side or at the server site to organize headlines for easy reading and to notify the updates to the end user. For real estate listings the RSS feed data consists of entries such as number of beds, number of baths, listing price, house type, address, and others.

Figure 7 shows the interaction between websites, the web server, the RSS reader, and the RSS feed. The circle with “1” represents the traditional approach and the circle with “2” represents the interaction using RSS feed. The diagram also shows that the browser first reads the content of Website1 and then that of Website2, but with the help of the RSS reader, the RSS feed for both websites is monitored simultaneously. Thus, any changes to multiple sites can be handled easily through the RSS feed.

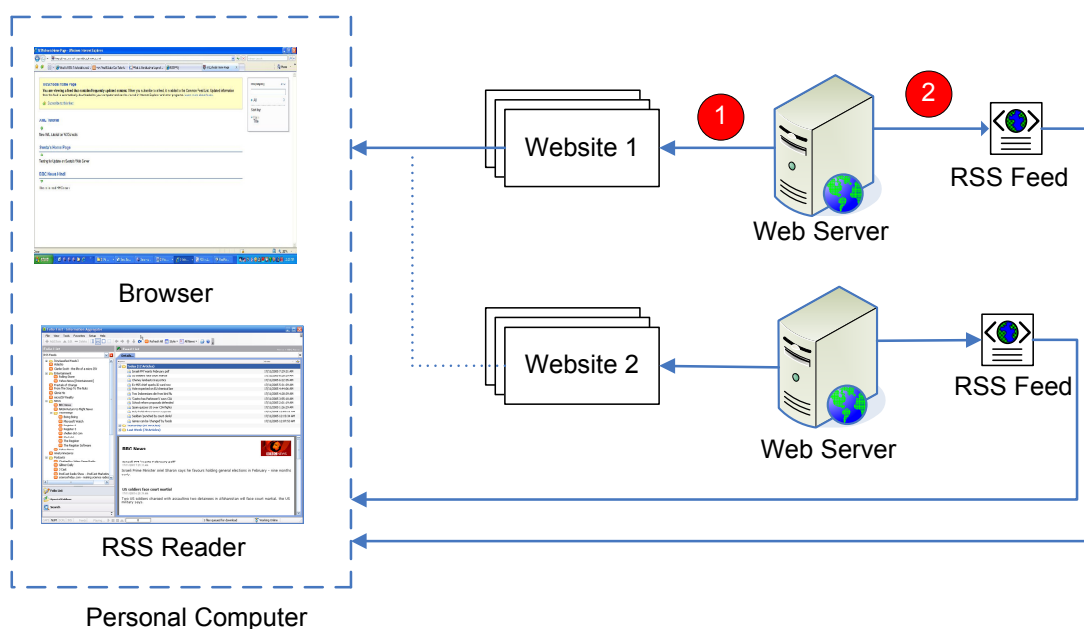


Figure 7. Interaction between Websites, Web server, RSS Reader, and RSS feed

RSS formats are XML based and uses tags in brackets to define its contents. The RSS feed data is contained in what is called “items.” The items are connected with some

theme or some similarity and these items are enclosed in a channel tag in the same way as an HTML tag. Each item contains title, description, and link. The title and description contain data to describe the contents and link is the link to the actual contents. The starting tags are enclosed in brackets and closing tags are enclosed with forward slash “/” similar to HTML usage. A sample RSS feed is shown in Table 6.

Table 6. Sample RSS Feed stream

```
<channel>
<title>The Channel Title Goes Here</title>
<description>The explanation of how the items are related goes
here</description>
<link>http://www.directoryoflinksgohere</link>
<item>
<title>The Title Goes Here</title>
<description>The description goes here</description>
<link>http://www.linkgoeshere.com</link>
</item>
</channel>
```

The RSS stream contains data in a standardized format, as shown in Table 6. Since the data is in a standard format, the web developer can read and interpret it and use it in other websites to populate their database of listings and automate the manual task of data entry that would help place other realtors’ or agents listings’ on the website. This would help drive more traffic on the website, getting more leads and more business.

Definition and Architecture of RSS Reader

RSS readers or RSS aggregators are software programs developed to access the RSS feed, interpret them, and organize the results for us. They are programs installed either on the client side or on the web server (e.g. Yahoo, News Gator), and Google Reader are readers available on the web and News Crawler and ROME are readers installed on user desktop computers.

ROME, which is the RSS reader used in Houses In Reach prototype, is an open source Java tool for parsing, generating, and publishing RSS data. It has the advantage that when we are coding in Java that we can extend the ROME Java libraries and implement them for Houses In Reach website when implementation is under a Java-based

environment. The ROME reader was built due to the lack of available of RSS parsers or generators that work with Java libraries.

The architecture of ROME is shown in Figure 8. **Architecture of ROME [Java2007]** The boxes represent classes in ROME APIs, the arrows represent the control flow between these classes, and the circles represent the sequence of the control flow.

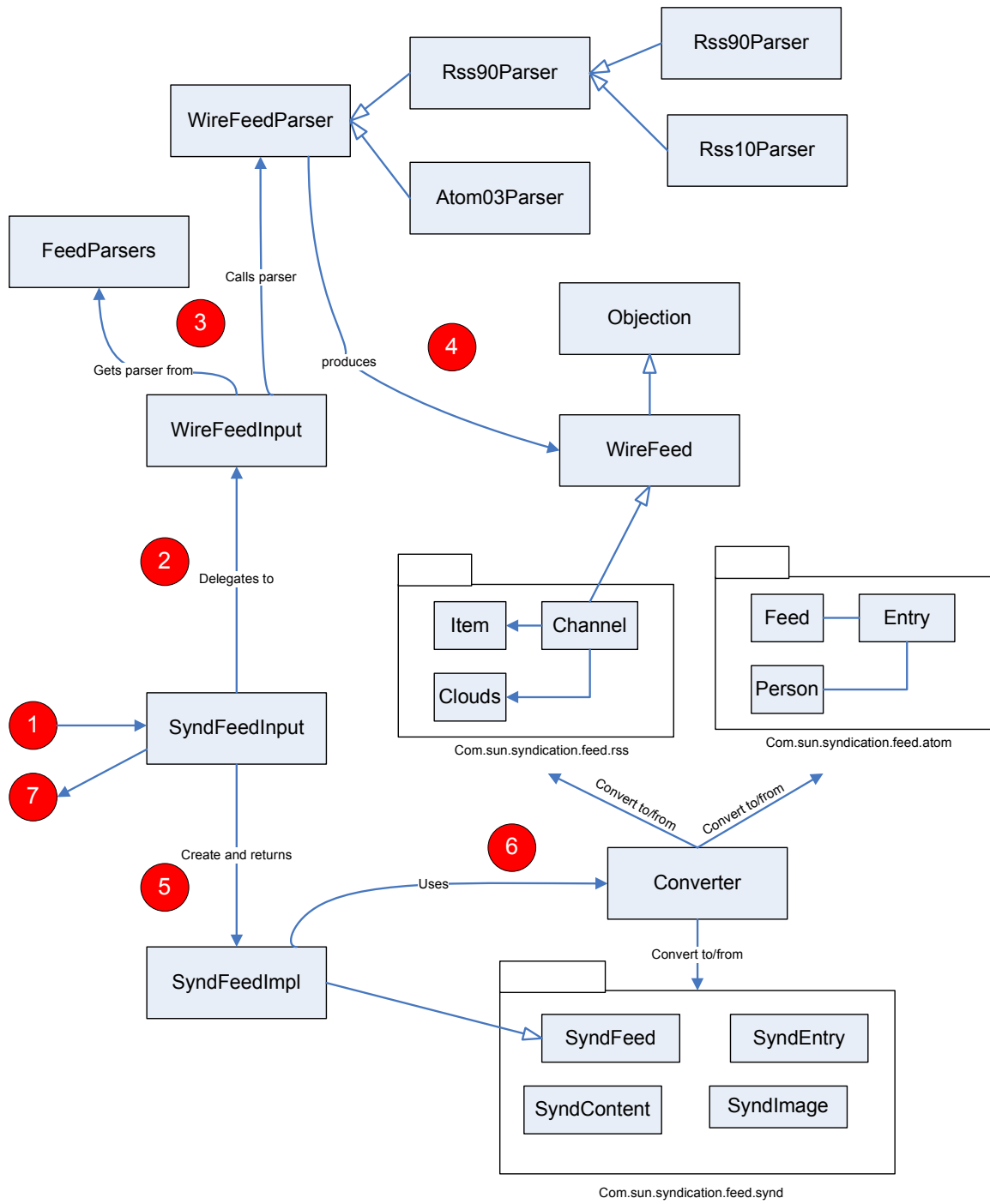


Figure 8. Architecture of ROME [Java2007]

Below are the steps RSS feeds go through when interpreted using ROME:

1. We call SyndFeedInput class to parse a feed as follows:

```
final URL feedUrl = new
URL("http://www.corcoran.com/rss/NewestPalmBeachesSale.xml");
final SyndFeedInput input = new SyndFeedInput();
final SyndFeed feed = input. build(new XmlReader(feedUrl));
```

2. The SyndFeedInput class delegates to WireFeedInput for parsing.
3. WireFeedInput uses a pluginManager class called FeedParses. This class is used to select the correct parser to parse the feed.
4. The appropriate parser parses the feed into WireFeed by using JDOM. In our case, the feed is an RSS feed, thus WireFeed is of class Channel and contains items from the com.sun.syndication.feed.rss package. In the end, WireFeedInput returns WireFeed.
5. The SyndFeedInput class uses WireFeedInput and creates a SyndFeedImpl that implements SyndFeed.
6. SyndFeedImpl uses a converter to convert between the format specific WireFeed representation and a format-independent SyndFeed.
7. SyndFeedInput returns a SyndFeed that contains the parsed feed.

3.4 Summary

Based on the supporting features of the real estate model presented by IMW, we designed Houses In Reach prototype to provide an effective environment for Houses In Reach users. We have described the Lindberg approach of investigating the role of consumer's belief value in the context of residential preferences and simulated choices. This chapter concluded with a description of the RSS feed and RSS reader and how they are useful for real estate websites. The notion of RSS has emerged as a standard for disseminating website contents to other web sites effectively. Both publishers and consumers are actively seeking ways to exploit this communication medium.

Chapter 4

System Architecture

This chapter describes the design of the Houses In Reach prototype real estate website. First, we present a model that corresponds to the Internet Media Works (IMW) model and close with a brief description of the major components of Houses In Reach prototype.

4.1 A model for real estate website applications

The introduction of e-commerce has improved business for realtors. The highest-earning realtors with a presence on the web achieve annual sales of four to eight million US dollars [Gaw2006]. However, the increase in the number of websites has also increased the competition among realtors. To survive in this competitive environment, realtors have to assist users efficiently, at lower prices. By automating various houses hunting processes within the enterprise, cost-effective websites can offers high quality services. Thus, the main goal of Houses In Reach website and our service is to promote, facilitate, and enhance the buying and selling of real estate by making access to and use of real estate listings more user-friendly, timely, and effective [Seth2003]. To achieve this goal, we must design an architecture that is flexible enough to upgrade to medium- or large-size applications. A suitable application architecture is depicted in Figure 9.

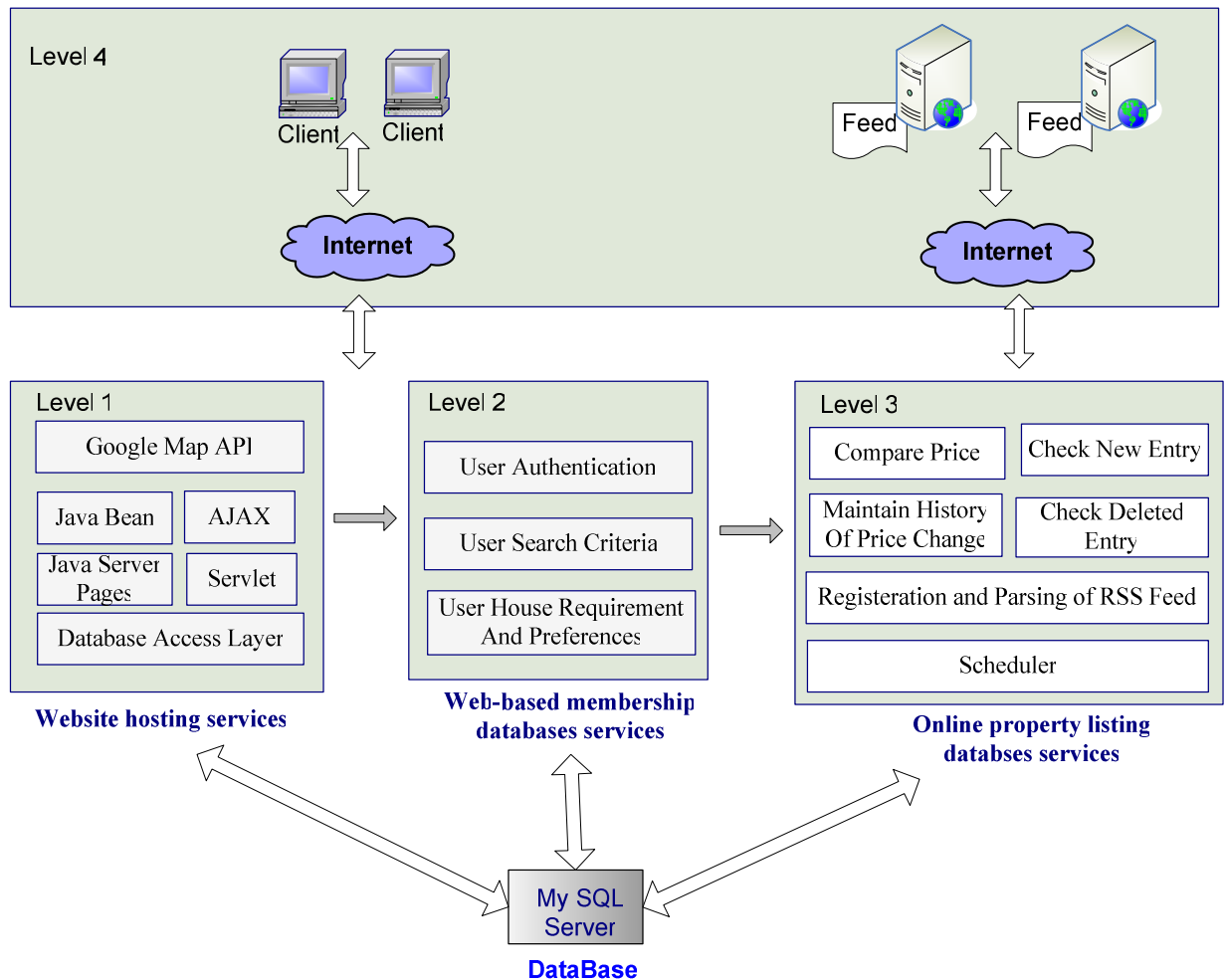


Figure 9. Four-Level Web-Based Real Estate Model

Figure 9 shows a four level data model compatible with the IMW [AnAA2004] data model for real estate as introduced in Chapter 3. The IMW as described earlier is specialized in web based application implementation and provides an architecture which helps sharing multi organization data from distributed websites. Thus adopting this model will help us to implement an effective website with easy design and support reusability. The following is the description of the adoption of various levels complying with the IMW model.

The first level consists of Google Map API, JSP (Java Server Pages), Java Bean, Servlet, and Ajax. This level contains the visualization engine component which

integrates Google Map API through JavaScript. The second layer holds the website users' personal information, desired house requirements, and their preferences for buying a house. This level is best fit for the user preference component model. This level also maintains different search names for different house requirements for a single user.

The third level holds and maintains the database for online property listings. This level supports the task of adding, deleting, and modifying property listings. In addition to this, this level will help to register and to parse the house information from the web and maintain a database of new house entries, deleted house entries, and changes in house prices; this will also track users affected by these changes. This task is done by a monitoring service that automatically manipulates online property listing data and the parsing engine to parse the data and save it to the database.

The fourth level of our application consists of the network services established at the client's location. These services include the database set-up, the web server set-up, and the set-up of websites implemented for real estate. These set-ups and installations are described in more detail in Appendix B.

4.2 Detailed component description for a real estate website application

This architecture presents a design with modularity, understandability, and reusability in mind. It supports modularity to facilitate the development of distributed and reusable modular elements. Here we describe in detail the structure of the first three levels of Houses In Reach website model. Level 1 provides a user-friendly and cost-effective website to the user. Level 2 provides access to a searchable database and controls user access to the website features. The function of Level 3 is to maintain databases for online property listings and provide additional features for the user to improve his or her e-business.

Level 1

Level 1 utilizes Ajax (Asynchronous JavaScript and XML), HTML, JSP, Servlet, AJAX and a database access. This Level enables serving dynamic web content to the user

and performing server-side processing of user requests. In order to serve dynamic contents and server-side processing, we use the strength of the Java Server Page (JSP) and Servlet technologies. Servlet is in charge of request processing and instantiating any Java beans to be used by JSP. JSP is responsible for retrieving any objects or beans created by Servlet and extracting dynamic contents for Servlet. JavaBean represents information on which the application operates.

In order to further describe Level 1, we have also used AJAX. The AJAX is a scripting technique in which the user performs some actions and sends a request to the web server. Instead of a browser loading a web page, as occurs in conventional web applications, the browser loads the Ajax engine, which is written in JavaScript. The Ajax engine is responsible for generating the interface for the user as well as for communicating with the server on the user's behalf. This communication is done asynchronously, which enables a paradigm shift on the web from merely receiving information to collaborating and dynamically combining data. Google Map is one such example of collaborating and dynamically combining data. Google Map enables the various derivatives that interactively combine data such as homes for sale or restaurants with map locations.

In order to present the collaborative communities and dynamic information-gathering and sharing, we will embed Google maps with JavaScript into our web pages. The use of this API enables the display of houses on the map to make aware of the locations of the houses. With the help of this API, we can incorporate Google map features, such as a display of brief house information in the Google information window. We can make the window interactive so that users can select houses for monitoring services.

As part of our model, access to a database is required. The database access level supports a database pooling feature in which data is retrieved from a database, bound into a database object, and sent to business logic. We have used the Java Database Connectivity (JDBC) API, which is the industry standard for database connectivity

between the Java programming language and a database, MYSQL in this application. It has a call-level API for SQL-based database access. JDBC technology enabled us to connect any data in a heterogeneous environment. The database design is available in Appendix C, Figure 42. **Database Design**

Level 2

The purpose of Level 2 is to create and maintain web-based membership databases and allow access to the searchable database. It creates and maintains a user database through user authentication. Establishing user authentication is possible for two types of users: a new user and a registered user. A registered user can access the website by entering his or her first name, last name, e-mail address, and a password. The system displays a message that the user is logged into the system. Login allows the user to create a new search, edit an old search, or view the status of previous listings. A new user enters the same login information and if the e-mail address and password have not been used previously, then the system creates a new account. The account for a new user is restricted to new searches. The system maintains the personal information of the user in a database. As the user is registered to the website or logged in, the system allows him or her to narrow searches by entering a preferred price range, style of property, age of the home, and other features of the property and naming those particular requirements as search criteria.

Narrowing of house searches by this level is satisfied by the user search criteria. The “search criteria” is the group name for house specifications. The system maintains these criteria so that the user can access them by specifying a name. For example, the user can provide “swe1 name” for houses with three bedrooms, two bathrooms, and so on. This facilitates specification of different searches for different requirements and does not limit the search to one set of requirements. The system maintains these house requirements for future monitoring purposes and sending updates to the user. After a user has selected the house requirements, he or she is directed to select the preference criteria.

Thus, house listings by this level are satisfied by selecting the user preferences criteria. The basic preferences criteria answer the question, “Why do I want to buy a house?” On the other hand, the preference criteria are derived from the twelve life values proposed by Lindberg [LiGM1989], and include cost, comfort level, proximity to health services, and so on (details of these preferences are discussed in Chapter 3 3.2 User preferences model). Depending on these choices, the system queries the database to display specific houses on the Google map. The system maintains user preferences in the same way it maintains house specifications for a user.

Level 3

Level 3 is the main channel through which we can maintain online property listings and provide housing updates to improve e-business. We can get online property listings through RSS feeds. RSS feeds provide a good resource for real estate data. In order to get the real time real estate data from RSS feeds, we need to register those feeds with the Houses In Reach website. The registration of RSS feed is done using a web page by entering and saving the RSS URL to the database. The website administrator, who is the user for this web page, is then allowed to enter a pattern of choices regarding fourteen items descriptive of the property. These fourteen items are: Bed, Bath, Title, Image, Description, Price From and Price To, Address, City, Province, Postal Code, Home Type, MLS ID, and Web ID. The patterns are for describing an item from the RSS feed.

The primary part of level 3 is the parser of the RSS feed, whose function is to read the RSS feed and save it to the local database of Houses In Reach website to perform future monitoring services for changes in price, new entries, and deleted entries. First-time parsing is saved into the main property listing table of the database and later parsing is done in an updated property listing table.

Level 3 also contains a means to compare prices. Comparison of prices starts after a new RSS feed is parsed and saved in the updated listing database. Price comparison is done between houses available in the main property listing table and the updated listing table. This level also checks new listings to recommend better choices to users. This can

be done using the “check new house” module. This module checks new house entries for the registered RSS feed. It starts after a price comparison is done.

Level 3 must also check any deleted entries in order to keep a user up-to-date with the house listing. This module checks the deleted house entries from the RSS feed. This level also contains a scheduler, which is a runnable thread that performs parsing, filtering, and checking of the updated entries on a prescribed schedule. This scheduler is responsible for managing the updates of houses. The sequence of the above modules is subject to change and can be executed depending on the preferences of the website users. The layer 4 is the networking services at client side.

The three levels of services are available in a core package. The first level describes the major technology for the real estate website. The second level maintains user records and the third level defines the implementation of features for real estate websites. The third level could have more features with other services for real estate websites.

4.3 Summary

This chapter has introduced the design rationale in brief for our real estate prototype website. Based on the supporting features of the real estate model presented by IMW, we designed a prototype named as “Houses In Reach” to provide a reusable component environment for our potential customers. The following chapters describe the implementation of this prototype in detail.

Chapter 5

The Visualization Engine

In our real estate application, we use a visualization engine, Google Map, for indicating the locations of houses. In this chapter, we describe the importance of the Google Map API and the basics of its use. Finally, we outline the Google Map API's limitations and difficulties.

5.1 Fundamentals of Google Map API

The web and online resources influence many aspects of life. Currently, this influence comes without a surprise as the web has changed the way we do business and search for information. The web is also able to deliver specific information about the locations of particular houses and allows a homebuyer to browse through information concerning the houses. This helps homebuyers examine choices that match their requirements more efficiently than the conventional method of inspecting in person each potential property. Thus, the depiction of the geographic location together with photographs or virtual tours of the property on the web can improve on-site visiting of a house. The Google Map API helps website developers integrate maps into their website.

Google Map is based on DHTML (Dynamic Hyper Text Mark-up Language), which means that a web page is different each time it is viewed, depending on different parameters. One can create DHTML in many ways, but Google Map API uses JavaScript. This API is a beta service available for any website and is free to the consumer. The developer requires a key to access the API. This key is available after registration of a website with the Google server. After registration, the developer must instantiate the Google Map class to integrate the map in a website, as explained in the next section.

5.2 Use of Google Map API for a website

Here, we walk the reader through the steps of how to integrate the Google Map API into a website. The integration of the Google Map API requires fourteen steps as of July 2007:

1. Register the key at the following URL:

```
http://www.Google.com/apis/maps/signup.html.
```

2. The following snippet loads the Google Map JavaScript, making it available for use in our JavaScript:

```
<script src
="http://maps.Google.com/maps?file=api&v=1&key=YOUR_KEY_HERE"
type="text/javascript"></script>.
```

This URL points to the location of the JavaScript file on the Google server that includes all of the symbols and definitions needed for using the Google Map API.

3. For displaying of a map, the developer needs to reserve a place for it using the div tag of HTML.

```
<div id="map" style="width: 800px; height:
500px"></div>.
```

We define a div name as “map” and set its size using the style attribute.

4. The GMap2 class is instantiated to define a map on the page in JavaScript:

```
var map = new GMap2(document.getElementById("map")).
```

5. To ensure that a map is placed on the page only after the page has fully loaded, we will execute the GMap2 function after the <body> element of the HTML tag receives an onload () event. This will help control how and when the map draws.

```
<body bgcolor="#FFFFFF" text="#000000" topmargin="2"
onload="load()" onunload="GUnload()">.
```

6. The developer initializes the map after creating it using GMap2. This is done by the setCenter () method. This method must be executed before any other operations are performed on the map:

```
map.setCenter(new GLatLng(37.4419, -122.1419), 13).
```

7. Next, the developer needs to refer to the locations on the map. This is done by retrieving a geocode address through predefined JavaScript commands. In order to do this, we invoke GClientGeocoder class:

```
var geocoderU = new GClientGeocoder().
```

8. Once invoked, the developer calls getLocation () method and passes a street address, which will retrieve the location information and caller’s status code. The response is passed to the callback function; that is, we obtain the latitude and longitude values

and place them on the map, which is `addAddressToMap ()`:

```
geocoderU.getLocations(userAddress[i],
    function addAddressToMap(response) {...}).
```

- To add the locations on the map, the object is first retrieved from the response containing the location inside the “response” object as follows:

```
var place = response.Placemark[0].
```

- Then retrieve a point for the map by passing the latitude coordinates followed by longitude coordinates and instantiating the `GLatLng` class, as follows:

```
var point = new GLatLng(place.Point.coordinates[1],
    place.Point.coordinates[0]);
```

- Create the marker on the map using the `createMarkerUser ()` function by passing the point as parameter, image, a counter, and the address of houses retrieved from the database after executing the query for user preferences, as described in Chapter 8:

```
1. var marker = createMarker(point,
2. '<table>
    <tr>
        <td colspan="1" align="right">
            <font size="2" face="Arial, Helvetica, sans-
                serif">Please check to select houses ---</font>
        <td colspan="2" align="center">
            <input type="checkbox" name="house'+id[j]+'>
            value="'+id[j]+'> onclick="selectMe(''+id[j]+'>"+
            tooltipText="Please Click to select this house"/>
        </td>
    </tr>'+
    <tr>
        <td colspan="1">'+</td>
        <td colspan="1" align="left">
            <font size="2" face="Arial, Helvetica, sans-
                serif">'+<b>Bed:</b>'+bed[j]+'<br>
            <b>Bath:</b>'+bath[j]+'<br><b>Price:</b>'+price[j]+
            '</font>
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <font size="2" face="Arial, Helvetica, sans-
                serif">'+<a href="'+link[j]+'> target="_blank"
                target="_top">'+address[j]+'</a></font>
        </td>
    </tr>
</table>'
```

```
3. , j, address[j]);
```

Line 1: Call function to create the marker by passing the location of that marker

Line 2: Pass the HTML code for that marker to display the houses in brief information window. The HTML code is for a check box to select the house, and an image in a window

Line 3: Pass the counter representing the element number and the address for the marker.

12. The description of creating a marker function is as follows:

```
1. function createMarkerUser(point,html,j,add) {
2.     var icon = new GIcon();
3.     icon.image =
4.         "http://localhost:8090/RealEstateNewVersion/image
5.         sNew/overlay/marker_34_red.png";
6.     icon.iconSize = new GSize(20, 35);
7.     icon.iconAnchor = new GPoint(6, 20);
8.     icon.infoWindowAnchor = new GPoint(5, 1);
9.     var d = gmarkers[0].getPoint().distanceFrom(gmarkers[1].
10.    getPoint())/1000;
11.     if ((d.toFixed(2))<50) {
12.         var marker = new GMarker(point,icon);
13.         GEvent.addListener(marker, "click", function(){
14.             marker.openInfoWindowHtml(html);
15.         });
16.     }
17.     return marker;
18. }
19. }
```

Line 2: Create the object for icon

Line 3: Load the image by specifying its location

Line 4: Set the size of the foreground image of the icon

Line 5: Set the pixel coordinate relative to the top left corner of the icon image at which this icon is anchored to the map

Line 6: Set the pixel coordinate relative to the top left corner of the icon image at which the information window is anchored to this icon

Line 7: Calculate the distance from the locations user has specified

Line 8: if the distance is less than 50 km, then go to step 8

- Line 9:** Create the object for the marker
- Line 10:** Create the event listener, which when clicked will open the information window containing the brief description of the houses
- Line 11:** Opens the map information window over the icon of the marker, containing the brief information when clicked by mouse
- Line 12:** Return the marker object
13. After creating the marker, add the marker control on the map as follows:
`map.addOverlay(marker) .`
14. Loop from step 8 to step 13, for every location.

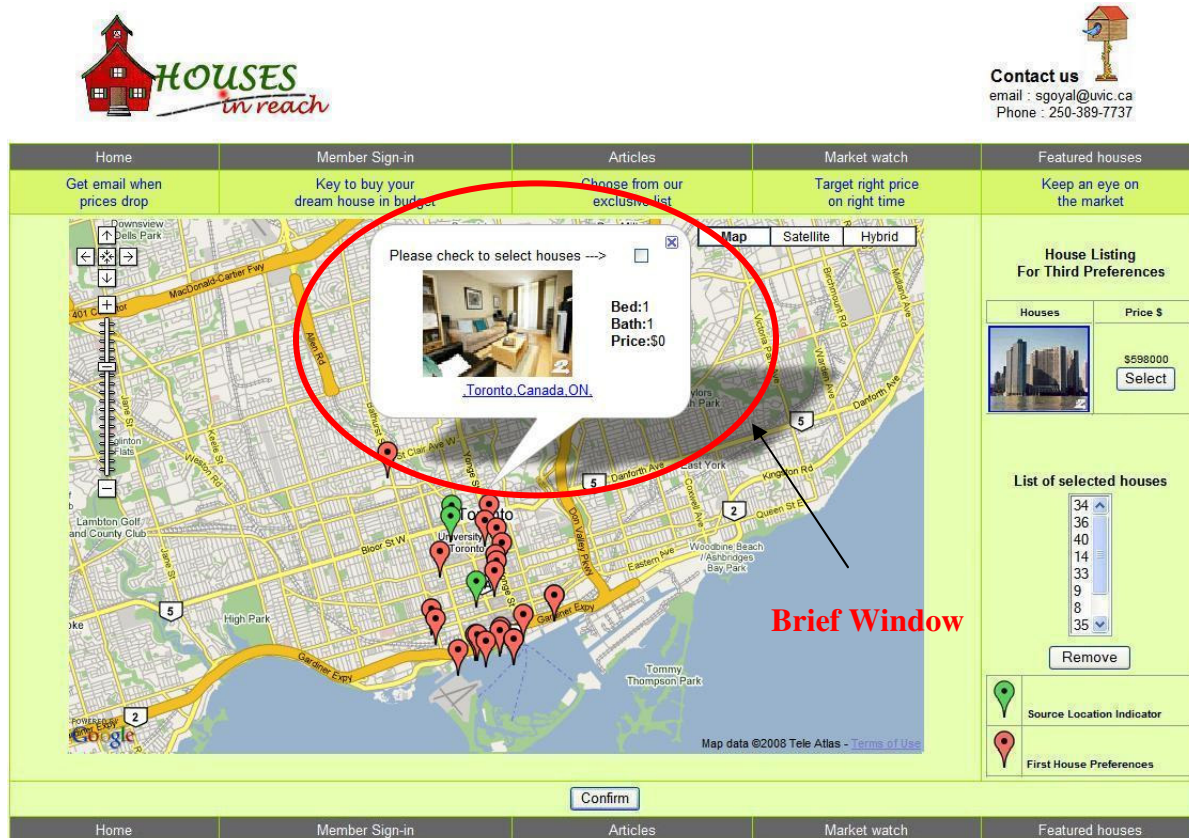


Figure 10. Sample Google Map with Information Window

These fourteen steps generate a web page as shown in Figure 10. This display helps to visualize the locations on the map that are close to user preferences. The green markers represent user preferences for recreation center, work place, and school. The red

markers locate relevant houses. The right hand side shows a list of identification numbers of houses selected by the user through clicking the check box available in the information window. This list is generated from the database.

The selection of houses is done through the selectMe () function available in step 11 (2) line. The detailed code of this function is shown here:

```

1. function selectMe(id){
    var LeftListBox =
      document.getElementById("SelectHouses");
2.   //alert (address);
3.   if (LeftListBox.options.length==0){
        //no item in the stack
4.     LeftListBox.options[0] = new Option(id);
5.   }else if ((LeftListBox.options.length==1)&&
        (LeftListBox.options[0].value=='title')){
6.     //Add the first house here...."
7.     LeftListBox.options[0] = null;
8.     LeftListBox.options[0] = new Option(id);
9.   } else{ //if items then set the flag as true
10.    var itemExist = false;
11.    for(var i=(LeftListBox.options.length -
        1);i>=0;i--){
12.      //find out if that id already exists
13.        if (LeftListBox.options[i].text==id){
14.          itemExist =true;
15.        } //end of if
16.      } //end of For
17.    if (itemExist){
18.      alert("Item already exists");
19.    }
20.    else{
        // create a new options
21.      LeftListBox.options[LeftListBox.options.length]=
        new Option(id);
22.      LeftListBox.options[LeftListBox.options.length].v
        alue = id;
23.      alert(LeftListBox.options[LeftListBox.options.len
        gth].value);
24.    }
25.  } //end of else
26. } //end of function

```

The above code first obtains the element's identification number and checks whether it is the first item or if items are already available in the list box. It also checks if the same identification number is already available in the list box. If it is not available,

then that identification number is added in the list box.

These steps generate a map depicting locations and a list of houses selected by the users of the website. This page is the main channel for the user for displaying houses and users can select houses for future reference.

5.3 Comparison with the Yahoo Map API

As well as Google, Yahoo offers Yahoo Map API. The Google API and the Yahoo API are very similar. We would go through similar steps to integrate a map in our web application using the Yahoo Map API. However, we have noted some basic differences between these APIs. Yahoo offers three APIs, whereas Google offers only one. Yahoo offers Yahoo! Map Simple API (developers can use this to put custom information on a Yahoo! Map without programming), Yahoo! Map Flash API (to embed maps in a website or application using Macromedia Flash Player) and Map Image API (to use DHTML and JavaScript to host self-developed maps).

Yahoo and Google both provide geocoding, that is, they both provide addresses and assess longitude and latitude and place a location on a map, using a marker. Other comparisons between the APIs are shown in Table 1.

Table 7. Comparison of Google and Yahoo Map APIs

Google Map API	Yahoo Map API
It provides geocoding from 2006	It provides geocoding long back then this
Maps can be generated for URLs restricted to the directory on our website	Maps can be generated only on a Yahoo site
It is a JavaScript API	It is XML (based on geoRSS 2.0) API embedded in JavaScript
It supports Ajax	It started to support Ajax from 2007
It has Zoom-In, Zoom-Out and Move-by-Mouse	It has Zoom-In, Zoom-Out and Move-by-clicking links
It can be used for commercial purposes, but should be freely available to end user	It can be used for commercial purposes, but developer should obtain written permission
It can add plain text or HTML or XML with XSL	It can display HTML in an IFRAME inside display window
It is still in beta version	It is not in beta and has a stable release
Usage is limited to upper bounds	Usage is non-restricted

Source: [1]

The Yahoo Map API offers faster performance and a smaller total download than the Google API. Regarding the details in satellite images, Google images are far better than those from Yahoo. Google Map API provides detailed images, including street views. However, we found a few limitations and difficulties during our implementations.

5.4 Limitations and difficulties

As of October 2006, after the introduction of geocode methods, Google Gadget's Google Map is easy to implement, requiring only one line of script. Its drawback is that it is not as customizable as the full API. Moreover, we found a few other limitations:

- Google Map API key is restricted to generating maps only for URLs associated with the website. We still need a local or remote web server to access the API for development.
- The API key applies only to a single directory or its subdirectory.
- Its usage is restricted to 50,000 hits per day.
- One is not allowed to obscure or delete any attributions or ads that appear within the map window.
- Whenever Google releases a new version of the API, the developer must update the site within a month to migrate to the new version.
- Google Map API is still in beta release; thus, it is entertaining to experiment with, but it is not for time critical work.
- Overlays are not pointed to the exact position after enlarging the maps.

Some difficulties in using the Google Map API are also apparent. Google Map API requires writing more code to achieve a simple task like placing different color overlays on maps than for example with Yahoo Map API. Writing the code becomes complicated when one must deal with different style markers for the same map.

5.5 Summary

In summary, assessing all important factors such as display of houses information, the Google Map API is the best API for our purposes. We have integrated the Google

Map interface to display houses on map and select them for monitoring services.

Chapter 6

Monitoring Utility

This chapter describes the monitoring capabilities of the website. We begin by describing the importance of monitoring for real estate websites. Subsequently, we describe the implementation of monitoring for the Houses In Reach website and close by discussing the limitations and difficulties of the service.

6.1 Importance of monitoring for a real estate website

Trading in real estate is very time-intensive. Buyers want to ensure that they are making the best possible choices. Most buyers using the web re-log in to the websites to keep track of house values, which is time consuming. Ideally a buyer would spend less time during the different stages of buying houses so that he or she can converge on the right choice by receiving information on only those houses that fit his or her needs. It is better if a buyer does not have to log in to every listing website and check for new listings, go through all of them, find interesting ones, and proceed with further investigation. It is also better if a buyer can see the history of price changes and keep track of houses that have sold. Hence, it would be more efficient if buyers were directed to a website that was maintained in such a way that they had only to log in and find all the new listings matching their preferences and a price trend to help making a decision on a house. This could be done by implementing a monitoring service to find new listings, keep track of price trends, and send alerts if a house for sale is no longer available in the real estate market. E-mailing the buyer about potential candidates when they arrive is also a good practice, as nearly 80% of web users check their mailboxes every hour [Inte2007].

The monitoring service is justified by a real scenario taken from a house buyer (see Appendix D). This scenario shows various problems with the conventional process of buying a house. The significant points of that scenario are as follows:

“When we started thinking, we had few concerns, like the state of the market. In terms of housing prices, we discovered that this is a buyers market, which could give us more negotiating power. And that’s the best time for buyers, when a lot of houses are available on the market and a

wide range of choices as well as prices as per our need.”

...

“So we started looking for “for sale” signs for all kinds of houses, single, new, old, and town houses. Again, we collected a lot of flyers and every day we came back and short listed some of the flyers. It was again overwhelming, since there were tons of houses on the market.”

...

“Once we decided, we short listed houses from our flyers and called them and visited them. We did it for months.”

...

“We kept an eye on some we short listed with our realtor, as well as looked more. Again, we went to see houses on our own and short listed a few of them.”

It is clear that the buyers wanted to know the market type by watching the price trends and wanted to keep an eye on the houses they had selected. They also wanted to have updates about their chosen houses. Sometimes, after considerable effort, they found that a house had sold or that the vendor no longer wanted to sell it. Hence, updates can save user time and energy. The process of updating and keeping an eye on houses for sale is referred to as monitoring, and can be performed in a variety of ways.

6.2 Monitoring for Houses In Reach website

To perform monitoring services on the Houses In Reach website, we add a feature that not only notifies the user of any changes through e-mail, but also lists new listings and other updates of preferences in the user account. Users are often interested in knowing the price history of a house to decide on the market trend, in receiving a list of new listings meeting their needs, and in discovering whether a price has changed and whether houses have sold.

A price heuristic provides an overview of the price trend. This is an excellent resource that allows a user to follow trends when buying a house. For example, the heuristic for three houses in a user’s list is shown in Figure 11. The data table at the top of the figure is the price history. The chart at the bottom of the figure is the representation of prices per unit from the date the three houses were made available in the database. The Y-axis is the price per unit and the X-axis represents the date of listing.

Historic Data

Houses	2007-10-13	2007-10-14	2007-10-15	2007-10-16	2007-10-17
House 1	\$135000	\$135000	\$125000	\$125000	\$125000
House 2	\$635280	\$635280	\$635280	\$635280	\$635280
House 3	\$479000	\$479000	\$479000	\$479000	\$479000

Heuristic

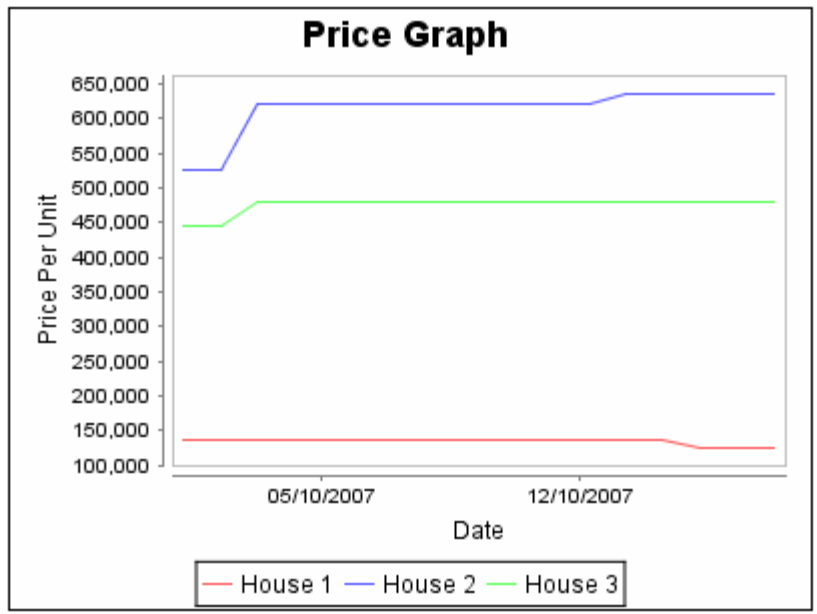


Figure 11. Record of Price History and Price Graph

To notify a user of new listings that meet user criteria, we e-mail the listings to the user as well as make the houses available in the user account. The user will then see the new updates when he or she next logs in to his or her account. The entry in his or her account is depicted in Figure 12; the user has the option to select a house by checking the check box. This will select the house and will update the user's house listing for future monitoring.

New Entries Matching Your Requirements

Houses	Bed	Bath	Price in \$	Address
	1	1	\$155000	51 Lower Simcoe St. Toronto, Canada, ON, <input type="checkbox"/>
	1	1	\$598000	46 Queens Quay Toronto, Canada, ON, <input type="checkbox"/>
	1	1	\$445000	126 Simco Street Toronto, Canada, ON, <input type="checkbox"/>

Figure 12. New House Entry in User Account

The next feature of our monitoring service is notification of a change in price. The purpose of notification is to inform a user that a house price has changed; he or she could then decide to buy the house. This notification is accomplished in two ways, through e-mail and through the display of the new price to user account, with the house listed as shown in Figure 13. The change in price can be seen in red in the current price column.

Listings of houses available

Houses	No. of Bed	No. of Bath	Current Price \$	Address
	1	1	\$659000	Toronto <input type="button" value="Remove"/>
	1	1	\$178800	Toronto <input type="button" value="Remove"/>
	1	1	\$140000	Toronto <input type="button" value="Remove"/>

Figure 13. The Change in Price Notation

Finally, in case a house is sold or is otherwise no longer available, then we can either e-mail the house status to the user or mark the house as sold, as shown in Figure 14. The “Remove” button allows the user to remove houses from his or her selection list. The price in red represents the change in price.

Listings of houses available

Houses	No. of Bed	No. of Bath	Current Price \$	Address
	4	5	\$859000	Toronto <input type="button" value="Remove"/>
	1	1	\$659000	Toronto <input type="button" value="Remove"/>
	2	2	\$598000	Toronto <input type="button" value="Remove"/>
	3	2	\$598000	Toronto <input type="button" value="Remove"/>
	1	1	\$199900	Toronto <input type="button" value="Remove"/>

Figure 14. The “Deleted” Notation

Monitoring is performed using a scheduler. The scheduler runs once or twice a day or on demand and notifies the user of any changes; it will also save changes to the database. Figure 15 depicts the complete Results page for a buyer.





Contact us
email : sgoyal@uvic.ca
Phone : 250-389-7737

Home	Member Sign-in	Articles	Market watch	Featured houses
Get email when prices drop	Key to buy your dream house in budget	Choose from our exclusive list	Target right price on right time	Keep an eye on the market

Results for your search on:

Listings of houses available

Houses	No. of Bed	No. of Bath	Current Price \$	Address
	2	3	\$125000	Toronto <input type="button" value="Remove"/>
	2	2	\$635280	Toronto <input type="button" value="Remove"/>
	2	2	\$479000	Toronto <input type="button" value="Remove"/>

New Entries Matching Your Requirements

Houses	Bed	Bath	Price in \$	Address
	1	1	\$155000	51 Lower Simcoe St. Toronto, Canada, ON, <input type="checkbox"/>
	1	1	\$598000	46 Queens Quay Toronto, Canada, ON, <input type="checkbox"/>
	1	1	\$445000	126 Simco Street Toronto, Canada, ON, <input type="checkbox"/>

Historic Data

Houses	2007-10-13	2007-10-14	2007-10-15	2007-10-16	2007-10-17
House 1	\$135000	\$135000	\$125000	\$125000	\$125000
House 2	\$635280	\$635280	\$635280	\$635280	\$635280
House 3	\$479000	\$479000	\$479000	\$479000	\$479000

Heuristic

Price Graph



Home	Member Sign-in	Articles	Market watch	Featured houses
------	----------------	----------	--------------	-----------------

Figure 15. The Results Page

6.3 The Limitations of Houses In Reach monitoring service

The monitoring service has a number of limitations.

- The monitoring service sends different e-mails for changes in price, for deleted entries, and for new entries.
- Unavailable houses will be shown as deleted, but there is no way to know if they are actually sold or taken off the market for other reasons.

6.4 Summary

The web could speed up the process of house-hunting by sending notifications of new offers, so that one can make decisions with less time and energy. This monitoring service is convenient and will encourage users to use the Houses In Reach website. Monitoring is required primarily for new housing entries, deleted entries, and prices changes.

Chapter 7

Parsing Engine

In this chapter, we describe registering, parsing, and saving RSS feeds to our local database. The parsing engine enables us to provide consolidated information to buyers and to provide a component for web developers to test their real estate implementation for real-time data. Initially, we describe the registration of a real estate RSS feed to Houses In Reach website. Subsequently, we describe the parsing component and saving parsed feeds to the database. Finally, we outline the limitations and difficulties of the parsing engine.

7.1 Registration of RSS feed

One important phenomenon of the Internet world is “Really Simple Syndication” or RSS. The RSS gives web users the ability to distribute contents and to make it widely available. The syndication makes contents available to not only blogs, online newspapers, and community and religious organizations, but to real estate websites. Many websites use syndication feeds to add house listings to a website. Syndication feeds are an efficient way to update house information and other information of interest such as real estate news to a website. They are easy to understand and every real estate agent can provide his or her own RSS feed. However, to take advantage of RSS feeds, a user has to go through each RSS feed, which is time-consuming. To overcome this problem, we can provide a website where users can obtain all the information consolidated from different real estate RSS feeds.

Even though RSS is easy to read and available with a standard format, different RSS feeds are available with different text formats, making it difficult for a web developer to consolidate all available feeds for a website. To alleviate this problem we have designed a form. This form is designed to support a level of indirection by referring the different text format RSS feeds items to the Java String functions and saving it to a database table. It helps to separate the data source which is the RSS feed with the programs operating on the data which is the monitoring service. Thus the data source and

the data integrator are independent to each other. The parsing engine helps to create that separation through this registration form and monitoring service is the program runs on the data source.



Contact us
 Email : sgoyal@uvic.ca
 Phone : 250-589-7737

Home	Create New RSS	Articles	Market watch	Featured houses
rss4		http://homes.point2.com/Rss.aspx?GeoArealD=290157		
Please Enter Your New RSS Feed :				
RSS Item : Bath		RSS Tag : Description		
RSS String Function : index e.g substring(), index()		RSS String Function Value : Bathroom(s) e.g Bed., Bath:		
RSS Index Start : -3 e.g 1 or 2		RSS Start Operation : Add		
RSS Index End : -1 e.g 1 or 2		RSS End Operation : Add		
RSS Character Value : NA e.g <,>		RSS Function Value : No		
RSS String Second Index : NA e.g index(), substring()		RSS String Second Index Value : NA e.g Bed., Bath:		
<input type="button" value="Test RSS"/>				
<pre> The Title is : \$239,900 CAD - Downtown, Toronto, ON, CA The Description is:
 Condo FOR SALE: 1 Bedroom(s) 1 Bathroom(s) The parserd value is:1 </pre>				
<input type="button" value="Submit New RSS"/>				
Home	Member Sign-in	Articles	Market watch	Featured houses

Figure 16. RSS Registration Form

Using the form, the web administrator enters a value that refers to a housing item. The different values entered in the form are the result of the questions available as follows:

- 1) Which item pattern are we defining? (*Form field name : RSS Item*)
For example, the item is “No. of Bedrooms.”
- 2) In which RSS tag is this item available? (*Form field name : RSS Tag*)
For example, the item “No. of Bedrooms” is available in <title> RSS tag.
- 3) What name is used in RSS feed to refer to that item? (*Form field name: RSS String Function Value*)
For example, the name “Bedroom” is used in RSS feed for “No. of Bedrooms.”
- 4) Which Java String function is needed to determine the position of the item in RSS feed? (*Form field name: RSS String function*)
For example, the index () Java String function is needed to determine the position of the item in RSS feed.
- 5) At what position does the value of that item start in RSS feed?(*Form field name: RSS Index Start*)
For example, in the pattern, “BedRoom: 3,” the value starts at position nine .The counting starts from the first occurrence of the pattern.
- 6) What operation do we need to extract an exact start position of the item value? (*Form field name: RSS Start Operation*)
For example, add the result of (4) and (5), or subtract the result for pattern “3 BedRoom(s).”
- 7) Do we have a character to determine the start the position of the item value in RSS feed? (*Form field name: RSS Character Value*)
For example, in a pattern like “Bedroom: 03
,” we have a character that helps determine the start position of the item value. We can loop to determine the length of the item value.
- 8) At what position does the value of the item end? (*Form field name: RSS Index End*)

For example, in the pattern “BedRoom: 3,” the value ends at position 10 or for the pattern “3 BedRoom(s),” the value ends at position 0.

- 9) What operation do we need to determine the end position of the item value?
(*Form field name: RSS End Operation*)

For example, add the result of (4) and (7), or subtract the result for pattern “3 BedRoom(s).”

- 10) Do we have a character to determine the end position of the item value in RSS feed? (*Form field name: RSS Function Value*)

For example, “Yes” for a pattern like “Bedroom: 03
,” we have a character that helps to determine the end position.

- 11) What Java String function is needed to determine the end character? (*Form field name: RSS String Second Index*)

For example, “index” in a pattern like “
 03 Bedroom(s)
 3 Bathroom(s),” we have a character that helps to determine the start position for a loop to ascertain the length of the character.

- 12) What is the end character value for that item value? (*Form field name: Rss String Second Index Value*)

For example, the end character for “BedRoom: 03
” is “<.”

If any of the operations have no value available, then the user enters “NA.”

The above questions will help register an RSS feed to the Houses In Reach website. After the feed is registered, we can parse and save the RSS feed. The form shown in

Figure 16 also allows users to test RSS patterns before saving them. Referring and saving RSS feeds in this way helps us to work with any real estate RSS feed. Hence, this form defines the level of indirection for the real estate RSS feeds to overcome working with different RSS text formats.

At first, we parsed a single RSS feed as depicted in Figure 17. The parser takes one RSS feed, parses it, and saves the results in the database. However, we quickly realized that for the personal real estate engine we needed the values of houses from

various websites and, therefore, we needed to parse a variety of different RSS feeds. The problem with the original approach was that it required the development of a new parser for every new RSS feed to be supported. While this approach is manageable for a developer, it does not work for an end-user of the Houses In Reach engine, who just wants to add another feed. However, we solved this problem by introducing a level of indirection as shown in Figure 18.

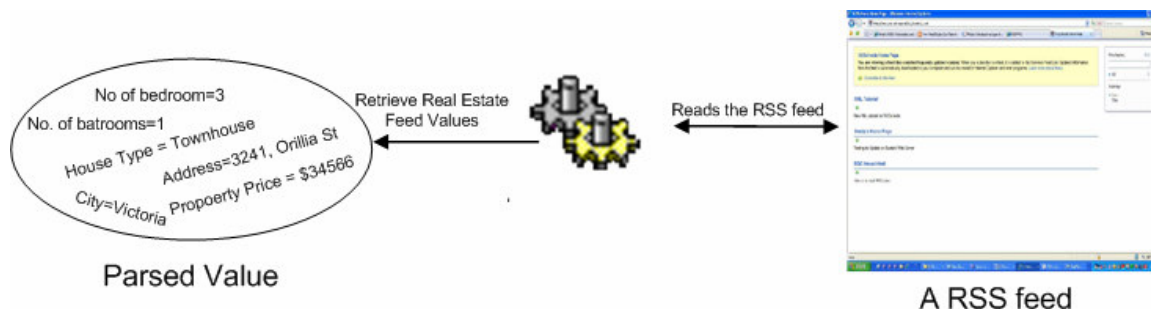
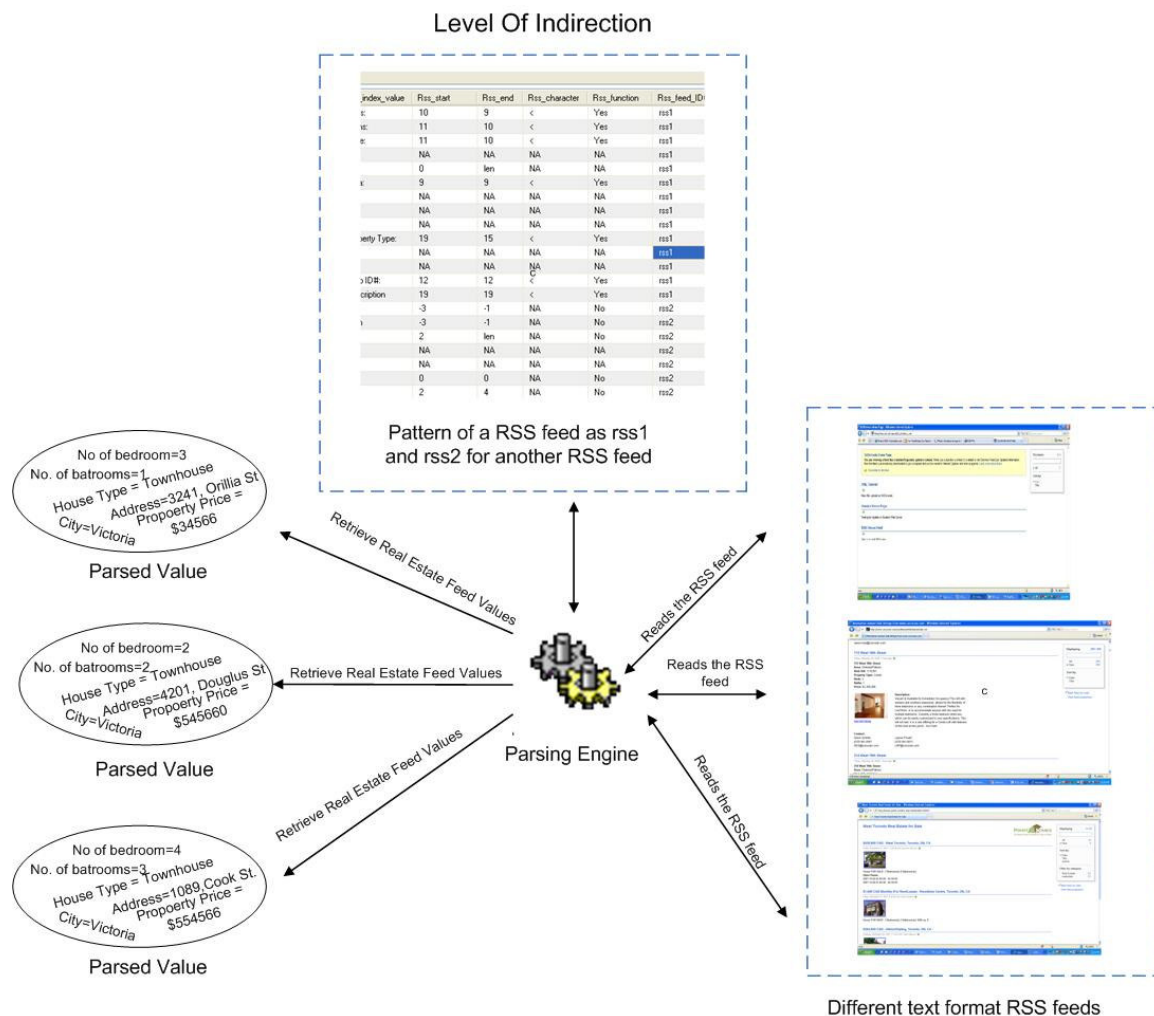


Figure 17. Direct RSS Feed Parsing Approach

To add a new RSS feed to the Houses In Reach engine, an end-user now specifies the structure of an RSS feed (i.e., the tag names and positions of the required data items) instead of writing a parser. The elicitation of the required information is guided by the RSS feed registration form depicted in Figure 17. This form and its corresponding database schema constitute a level of indirection. The design of this form was driven by the patterns of existing RSS feeds and the needs of Houses In Reach engine. To automate the process of adding a new RSS feed further, we save a number of registration form descriptions in the database for future reference. This user-friendly RSS feed parsing approach, which uses a level of indirection, is depicted in Figure 17 below. RSS feed patterns stored in the database can also easily be updated using the RSS shown in Figure 15.

This level of indirection, which features a intermediate data description, is a common software engineering approach to facilitate data interchange from different sources.

The implementation of the parsing engine is described in next section.



7.2 Detail of Parsing Engine

The parsing engine reads a RSS feed from any real estate RSS feed, parses it, and saves it in the database. The component reads RSS feeds with the help of ROME, an open source java API, parses it using the utility described below, and then saves it in a database based on MYSQL. This component follows the steps described here in this section. We present these steps in a few lines of java code, followed by their explanation

and a sample to explain how the component performs in real time. This sample code parses the number of bedrooms from an RSS feed.

The monitoring service performs a number of steps to parse the RSS feed:

Step 1: Select the URL of the registered feeds from the database to be parsed:

```
1. GetFieldsFromTable fieldFromTable = new
   GetFieldsFromTable()
2. ArrayList urlList = fieldFromTable.getRows("Rss_Feed_URL"
   "Rss_Feed_TableName", "rss_feed")
3. for (int i=0; i<urlList.size();i=i+2){...}.
```

Line 1: Create the instance of GetFieldsFromTable class to retrieve the rows from the table.

Line 2: The getRows () method retrieves the values and returns it to the array list. This method has three parameters: "RSS_Feed_URL" and "RSS_Feed_TableName" are the field names available in the database and "rss_feed" is the name of the table that contains all registered feeds.

Line 3: Loop to parse all RSS feeds registered for Houses In Reach website.

Example

The code retrieves the registered URL for RSS feed:

```
urlList = http://www.corcoran.com/rss/NewestManhattanSale.xml.
```

This feed is available on the browser with the house listing in Figure 19.

14 East 90th Street

Wednesday, December 19, 2007, 8:26:25 PM →

14 East 90th Street**Area:** Carnegie Hill**Web ID#:** 1141464**Property Type:** Coop**Beds:** 3**Baths:** 3**Price:** \$3,500,000[view full listing](#)**Description**

A TRUE DIAMOND IN THE ROUGH! Create your own masterpiece in this spacious classic 7 room apartment in estate condition in one of the best buildings in town. Located in Carnegie Hill just a few steps from Central Park, incredible shopping, restaurants, Museums, etc. 14 East 90th proudly boasts a gym, a magical childrens playroom and a mini basketball court. There is a large, caged storage bin that conveys with the unit as well. Bring your imagination and your architect/designer.

Contact:

Sheila Wyle, VP

(212) 360-2259

SFW@Corcoran.com

Figure 19. A Sample RSS Feed

The XML for the feed of Figure 19 is shown in Table 8.

Table 8. The XML for the Feed of Figure 17

```

<item>
<title>
xmlns:cf="http://www.microsoft.com/schemas/rss/core/2005"
cf:type="text">14 East 90th Street</title>
<link>http://www.corcoran.com/property/listing.aspx?Region=NYC
&amp;ListingID=1141464</link>
<description>
xmlns:cf="http://www.microsoft.com/schemas/rss/core/2005"
cf:type="html">&lt;b>14 East 90th
Street</b>&lt;br>&lt;b>Area:</b> Carnegie
Hill&lt;br>&lt;b>Web ID#:</b>
1141464&lt;br>&lt;b>Property Type:</b>
Coop&lt;br>&lt;b>Beds:</b>
3&lt;br>&lt;b>Baths:</b>
3&lt;br>&lt;b>Price:</b>
$3,500,000&lt;br>&lt;table border=0 cellpadding=0
cellspacing=0 width=548&gt;&lt;tr&gt;&lt;td width=199
valign=top&gt;&lt;img

```

```

src="http://www.corcoran.com/images/media/UnitPhotos/475036.1.
jpg" width=160 height=120 border=0><br><br>a
href="http://www.corcoran.com/property/listing.aspx?Region=NYC
&amp;ListingID=1141464">view full
listing</a></td></td
valign=top><b>Description</b><br>A TRUE
DIAMOND IN THE ROUGH! Create your own masterpiece in this
spacious classic 7 room apartment in estate condition in one
of the best buildings in town. Located in Carnegie Hill just
a few steps from Central Park, incredible shopping,
restaurants, Museums, etc. 14 East 90th proudly boasts a gym,
a magical children's playroom and a mini basketball court.
There is a large, caged storage bin that conveys with the unit
as well. Bring your imagination and your
architect/designer.</td></tr></tr></td
width=548 valign=top
colspan=2><br><b>Contact:</b></td>&
lt;/tr></tr></td width=199 valign=top>Sheila
Wyle, VP<br>(212) 360-
2259<br>SFW@Corcoran.com</td></td
valign=top></td></tr></table></descriptio
n>
</item>

```

Step 2: Parse the syndication feed using ROME. The syndication feed in ROME is handled by the WireFeedInput class. The WireFeedInput class parses an XML document into WireFeed by choosing the correct parser using a pluginManager class called FeedParser as discussed in Chapter 3 , Section 3.3 RSS Feeds and RSS Reader for real estate website applications. Parsing using ROME can be performed as follows:

1. WireFeedInput wInput = new WireFeedInput();
2. WireFeed wireFeed = wInput.build(new XmlReader(feedUrl));
3. Channel channel = (Channel)wireFeed;
4. items = channel.getItems();
5. for (Iterator iter1 = items.iterator(); iter1.hasNext();){
6. Item fieldValue = (Item)iter1.next(); ...}

Line 1: Create an instance of the WireFeedInput class.

Line 2: Read the syndication feed through a build () method by passing the RSS feed URL obtained in Step 1.

Line 3: WireFeed is of class Channel, which is a Java Bean for RSS and stores all the RSS information tags.

Line 4: Return the channel items containing the list of items from the RSS.

Line 5: Iterate on each house listed in the RSS feed.

Example

Table 9. The Value Parsed by ROME

<pre>The FieldValue: Item.title=14 East 90th Street Item.content=null Item.categories=[] Item.link=http://www.corcoran.com/property/listing .aspx?Region=NYC&ListingID=1147795 Item.description.value= {Shown in Step 3}</pre>
--

Step 3: Select the RSS feed pattern saved for that feed while registering it:

1. `ArrayList allRssFeaturesRow = fieldFromTable.getRows("rss_features", "Rss_feed_ID = '"+rssFieldValue+"'");`
2. `for (int j = 0; j<allRssFeaturesRow.size(); j=j+14){.....}`

Line 1: The `getRows ()` function retrieves the feed pattern for that feed from the database. "rss_features" is the name of the table and "rssFieldValue" is the identifier for the RSS with which we are working and "allRssFeaturesRow" is the array list containing all rss pattern items.

Line 2: The "for" loop loops over each item in a single house list, for example, the number of baths or bedrooms.

Example

Table 10 shows the database field name and the patterns as form field values. The `allRssFeaturesRow` contains these patterns for all fourteen available items.

Table 10. Database Field Name and Value

Form Field	Form Field Value	Field Name
	1	Rss_ID (primary key)
RSS Item	Bed	Rss_item
RSS Tag	Description	Rss_tag
RSS String Function	index	Rss_index
RSS String Function Value	Beds:	Rss_index_value
RSS Index Start	10	Rss_start
RSS Index End	9	Rss_end
RSS Character Value	<	Rss_character

RSS Function Value	Yes	Rss_function
RSS Feed Name	rss1	Rss_feed_ID
RSS Start Operation	Add	Rss_Start_Operation
RSS End Operation	Add	Rss_End_Operation
RSS String Second Index	NA	Rss_index2
RSS String Second Index Value	NA	Rss_index2_value

Step 4: Next, we determine the RSS tag where that item value exists. To do so, we retrieve the pattern that defines that RSS tag from the database table and call the appropriate method for that tag value:

```

1. rssTagDescript = (String)allRssFeaturesRow.get(j+2);
2. if (rssTagDescript.equals("Item"))
3.     validateItem(fieldValue);
4. if (rssTagDescript.equals("Title"))
5.     parseTitle(allRssFeaturesRow, rssFieldValue, j);
6. else if (rssTagDescript.equals("Description"))
7.     parseDescription(allRssFeaturesRow,
8.         rssFieldValue, j);
9. else if (rssTagDescript.equals("Category"))
10.    parseCategory(allRssFeaturesRow,
11.        fieldValue, fieldFromTable, rssItemRows, rssField
12.        Value, j);

```

Line 1: Get the RSS tag name for the pattern from the arrayList obtained in Step 2.

Line 2: If the table says that item is in item tag of RSS, then

Line 3: validate the availability of items in “title” tag or in the “item” tag. fieldValue contains the value of the item retrieved in step 2, line 6.

Line 4: If the table shows item is in “title” tag, then

Line 5: parse the “title” tag.

Line 6: If the table shows item in “description” tag, then

Line 7: parse the “description” tag.

Line 8: If the table describes item in “category” tag, then

Line 9: parse the “category” tag.

Example

From Table 10, the RSS tag is “Description.” This implies that the number of bedrooms is available in the description tag, as shown in Table 11.

Table 11. The Value of Description Tag

```

rssTagDesc  =<b>14 East 90th Street</b>
<br><b>Area:</b> Carnegie Hill<br>
<b>Web ID#:</b> 1141464<br><b>Property Type:</b> Coop<br>
<b>Beds:</b> 3<br>
<b>Baths:</b> 3<br>
<b>Price:</b> $3,500,000<br>
<br>
<table border='0' cellpadding='0' cellspacing='0' width='548'>
  <tr>
    <td width='199' valign='top'>
      <img
src='http://www.corcoran.com/images/media/UnitPhotos/475036.1.jpg
' width='160' height='120' border='0'><br>
<a
href='http://www.corcoran.com/property/listing.aspx?Region=NYC&Li
stingID=1141464' target='_blank'>view full listing</a>
    </td>
    <td valign='top'>
      <b>Description</b>
      <br>A TRUE DIAMOND IN THE ROUGH! Create your own masterpiece
in this spacious classic 7 room apartment in estate condition in
one of the best buildings in town. Located in Carnegie Hill just
a few steps from Central Park, incredible shopping, restaurants,
Museums, etc. 14 East 90th proudly boasts a gym, a magical
children's playroom and a mini basketball court. There is a
large, caged storage bin that conveys with the unit as well.
Bring your imagination and your architect/designer.
    </td></tr>
    <tr><td width='548' valign='top'
colspan='2'><br><b>Contact:</b></td></tr><tr><td width='199'
valign='top'>Sheila Wyle, VP<br>(212) 360-
2259<br>SFW@Corcoran.com</td><td valign='top'>
    </td>
  </tr>
</table>

```

Step 5: Perform the operations as defined in the pattern to retrieve the value. This operation is saved as a pattern during registration of the RSS feed. This can be done using the following function (described in detail in Appendix E):

```

getRssValue(String rssTagDesc, List allRssFeaturesRow,
int j ) {...}.

```

The “rssTagDesc” is the tag description, the “allRssFeaturesRow” is the list containing all the RSS feature values for that item from the table, “j” is the integer value determining the position in the array to retrieve the RSS features saved, and rss is the

flag.

Example

The operation retrieves the number of bedrooms from the RSS feed using the above function. The number of bedroom fields exists in the Description tag, as shown in bold in Table 11. We use an index Java String function for extracting the position of the pattern “Beds:”. This function returns the first occurrence of “Beds:” in the description tag. For example,

```
IndexValue = rssTagDesc.indexOf("Beds:")
```

results in the IndexValue = 120.

We use a substring (start, end), which is a Java string function, to extract the value of the number of bedrooms. First, we obtain the start position of the item value. We have the position of “Beds:” pattern as an IndexValue. Now adding the value of “Rss_start” as available in Table 10, to the IndexValue gives us the actual position of the item value. Second, we determine the end position of the item value, which is more difficult. The value of the number of bedrooms can be no value (Beds=Not Available), one digit (Beds=1), or two digits (Beds=02). However, looking at the pattern in the description, the number of bedrooms is always followed by the “<” character. We can determine the number of digits by starting a loop from the value of “end” as the rss index end forms a field until it hits the “<” character. The character “<” in the form specifies “Rss_character.” This gives us the actual length of the item value. Now adding that length to the “Rss_end” value available in Table 10 determines the “end” for the substring.

Thus the start =10

End=9

```
rssEachValue[arrayIndex]=rssTagDesc.substring(IndexValue +start,  
IndexValue +end).trim()
```

rssEachValue[arrayIndex]= **3** which is the number of bedrooms specified in RSS feed as available in Table 11.

The index value and start values are added together because the “Rss_Start_Operation” and “Rss_End_Operation” are ADD in Table 10.

The Rss Function Value implies that the item is available in the RSS feed. For example, if the rss function is “Yes” for number of bedrooms, then the description tag contains the number of bedrooms, but if the rss function is “NA,” then the rss function value is not available for the number of bedrooms item.

Step 6: Save the result in the database.

```
1. SaveSellers saveSellers = new SaveSellers();
2. int noRows = saveSellers.insertSellersValue(con, rssValue,
    tableName);
```

Line 1: Instantiate SaveSellers class to save RSS values in the database.

Line 2: InsertSellersValue () saves the values in the database. The “con” is the Connection object and “rssValue” is the ArrayList containing values and “tableName” is the name of the table. The code is available in Appendix E.

7.3 Difficulties and Limitations

The RSS feed process has a number of difficulties and limitations.

- The RSS feed sometimes does not have the correct information. For example, number of bedrooms available may be erroneously listed as 0.
- The prices value is not standardised. In real estate, \$23 may mean \$23,000 or \$230,000. Thus, some RSS feeds follow an abbreviation and others list a price.
- A house is considered sold if it is unlisted from a RSS feed. No explicit value distinguishes a house that is actually sold from one that is merely unlisted.
- No standard identifier distinguishes houses. Thus, the fact that the same house is listed in different RSS feeds is not recognized.

7.4 Summary

Despite the limitations, the RSS parsing engine could parse any real estate RSS feed. This parsing component is useful for buyers, since it provides a consolidated listing

of properties available on the web. It is also useful to the web developer, as it allows the web developer to test a real estate website with data tested in live environment. The parsing engine works as a channel to provide opportunity for venders to sell the houses listed on their website.

Chapter 8

User Preferences

This chapter describes residential preferences and choices a buyer could make while searching for properties using the Houses In Reach website. After describing the user preferences data model proposed for the Houses In Reach website, which helps retrieve houses based on user preferences, we discuss the algorithm used for the implementation. The user preferences model used in the Houses In Reach engine is based on the model proposed by Lindberg et al. [LiGM1989]. We end with a description of the forms used to implement the data model and the results of the algorithm. To the best of our knowledge, this is the first implementation that actually applies the model proposed by Lindberg et al.

8.1 Data model for real estate website for user preferences

As the numbers of users who search for houses online increases daily, it becomes increasingly important to develop a web-based product to cater to the various requirements of online users and help them buy a house efficiently. To do so, we must disentangle complex residential choices and preferences. We do this by proposing a data model and then implementing it on a website.

Our choice of user preferences is justified by a real time scenario taken from a house buyer (see Appendix D). This scenario shows various problems in the conventional process of buying a house. The significant points of that scenario follow:

“We had a wide variety to choose from. There were condos available in the city, but privacy was lacking, most were a little far from the city, but had good prices; many were in 2-3 story high rises. We also looked at luxury condos, posh locations, nice community, good facility, some with personalized offers be ready in future, some on seasonal discounts, also on sale by owners (pre-owned). We thought we’d start by looking at pre-owned condos since they might be more reasonable. So we started looking around in the city for “on-sale” boards, we searched the availability on the internet, we visited realtor’s sites, and we saw ads in papers and flyers, and started looking and talking to owners.”

...

“It took a few months for us, and during this process we somehow felt that

owning a condo is almost like living in an apartment, only it's ours.”

...

“So we started looking for “for sale” signs for all kinds of houses, single, new, old, and town houses.”

We see that a buyer cannot know beforehand what his or her choices are. He or she starts with a broad perspective and with some research makes decisions and pick a few houses close to his or her preferences. It may take weeks to determine exactly what a buyer wants. Thus, we need an approach that determines houses close to user requirements and lists them as suggestions. This not only saves user time, but also helps him or her to converge on the best possible house. Lindberg *et al.* [LiGM1989] has proposed three models and computed an index that shows the perceived effect of housing attributes on the attainment of life values available in Chapter 3 Table 5. We advance that index as our data model. The benefit of using Lindberg's approach is that it is quantitative and based on a user study, which makes it easy to order user preferences at different levels; it can also be used to determine house listings close to user requirements.

We categorize the ratings from the index table at three levels. The first level includes values between 4.1 and 6.0. The second level includes values between 2.1 and 4.0. The third contains values less than 2.0. This categorization will help buyers to evaluate properties from best (6) to worst (-6) choices. The data model we devised is depicted in Table 12. The life values are available in the first column. The housing attributes are priorities based on ratings.

Table 12. User Preferences Data Model [LiGM1989]

Life Values	First Preference 4.1 - >6.0	Second Preference 2.1 - 4.0	Third Preference -6.0 - 2.0
Comfort	Size-5.50 Recreation-4.08 Work-4.36	Standard-3.83 Downtown-3.36 Schools-3.42	Cost-1.14
Excitement	Recreation-4.19	Cost-3.28 Size-2.03 Downtown-2.17 Work-2.36	Standard-0.25 Schools-1.89
Family	Size-6.06	Cost-2.39 Standard-2.17	Downtown-1.00

		Recreation-3.03 Schools-3.94 Work-3.33	
Freedom	Size-4.52	Cost-2.75 Downtown-2.44 Friends-2.69 Schools-3.92 Work-4.00	Standard-1.22
Happiness		Size-2.86	Cost-1.58 Standard-0.86 Downtown-0.53 Recreation-1.92 Schools-1.36 Work-1.42
Health		Size-2.39 Recreation-3.11	Cost-0.86 Standard-0.72 Schools-1.08 Work-1.42 Downtown- -1.0
Inner harmony		Size-3.69 Recreation-2.11 Schools-2.19 Work-2.39	Cost-1.75 Standard-0.67 Downtown-0.69
Leisure	Recreation-5.28	Cost-2.22 Size-2.08 Work-2.53	Standard-0.25 Downtown-1.31 Schools-1.89
Money	Cost-6.22 Work-3.47	Size-2.63 Standard-2.22 Schools-2.17	Downtown-1.92 Recreation-2.00
Pleasure		Size-2.78 Recreation-2.03	Cost-1.22 Standard-0.75 Downtown—0.19 Schools-1.11 Work-1.42
Security		Schools-2.44	Cost-1.75 Size-1.78 Standard-1.00 Downtown-0.42 Recreation-1.33 Work-1.89
Togetherness		Size-3.19 Recreation-3.06 Schools-2.36 Work-2.25	Cost-1.06 Standard-0.97 Downtown-1.44

Table 12 shows that if a user has comfort as his or her chosen life value, then

houses that have certain sizes, a range of distances from recreational facilities, and a range of distances from work places should be rated as first preferences. Second preference should be given to houses close to downtown and schools, and other standard specifications. Third preference should be given to houses that fall in the user price range. The first preference might contain a house that costs less than or more than the standard requirement.

The system picks the value for the first preference and retrieves the house values from the database for the life value selected by the user. The system continues with the next level. The houses are filtered for replicate values and all the filtered houses are sent for display either on the map or on the side of the map depending on whether the house has a location in its preference.

Neighbourhood attributes are not considered as the part of this data model because these attributes require a third-party data source or data in RSS feed. By using the model devised in Table 12, the Houses In Reach application will suggest houses to help buyers in making decisions; instead of having to sort through both relevant and irrelevant houses, users will see houses close to their preferences. Our model displays houses based on user preferences on a map. This implementation is done using an algorithm, which is explained in the next section.

8.2 User preferences algorithm

An algorithm for the Lindberg *et al.* index of perceived effects of housing attributes on life values generates a dynamic query to obtain the houses from a database created after saving RSS feed values locally [LiGM1989]. The steps of the algorithm are as follows:

Step 1: Obtain user life values from the user's form:

```
userPrefItem = userLoginBean.getUserPrefName();
```

Step 2: Depending on user life values, the system obtains the house attributes:

```
public ArrayList getUserPrefFieldValue(String
    userPrefItem, String prefType) {
```

```

1:   int IndexLV = getIndexLV(userPrefItem);
2:   String[] ValueHA = getRatings(IndexLV,prefType);
3:   fieldValue = getUserFieldValues(ValueHA);
       return fieldValue;
};

```

Line 1: The `getIndexLV()` retrieves the life value from the array.

Line 2: The `getRatings()` retrieves the housing attributes from the array based on the `indexLV` and `PrefType`. `PrefType` is the preference type obtained from the user form. The advantage of getting the housing attributes from a array is that it can further extend for different results, if evolution of data would occur. The detail of the user preferences code is available in Appendix F.

Line 3: The `fieldValue` is the `ArrayList` containing house attributes for the `prefType` which is the first, second, or third preference and the `userPrefItem` obtained in Step 1.

Step 3: Query the database with house attributes and user value for that house attribute and retrieve the first preference:

```

ArrayList propertyRowValue =
getTableRowValue(pref[i],propertyFieldValue,propWhereClause
);

```

The `userFieldValue` is the `ArrayList` obtained in Step 2. `pref[i]` is the array for three levels. The `propertyFieldValue` is the table field obtained with respect to the housing attributes. `propWhereClause` prepares the “where” clause to retrieve the rows from the database. This clause can be created using the following function:

```

public ArrayList getPropertyFieldValues(ArrayList
userFieldValue){...}.

```

Step 4: Find if house attributes have locations. If they have locations, then the algorithm must create a response string to display the location on a map.

```

1. String charArray = checkLocation(userFieldValue);
2. userDataBean.setprefChar(charArray);
3. CompareAndPopulatePref(propertyRowValue,pref[i]);

```

```
4. responseStr =
    createResponseString((ArrayList)propRowMap.get(index), propertyFieldValue);
```

Line 1: Check if the housing attributes obtained in Step 2 have a location.

Line 2: Compare and populate the row values; however, if we are evaluating first preference, then there is nothing to compare. If the preference is second or third, the program checks if the house already exists; if it exists, then the program removes it from the lower level preferences; otherwise it creates a string for the second preference.

Line 3: Line 3 creates a response string to send to the client to display houses in a map. The propRowMap is an arraylist that is opened in the CompareAndPopulatePref function containing the list of all the houses.

Step 5: If there is no map and a response string for display on the side of the map is created, then

```
propRowNonMap.add(indexNoMap, propertyRowValue);
```

Step 7: Repeat Step 2 and Step 6, to create the response string for the second and third preferences.

The response string is sent to the client browser, where the browser calculates the distance from user-specified recreational centre addresses, a work place address and school address, and displays houses on the map. However, the values that function as input for this algorithm are obtained from the three forms and the algorithm results are explained in next section.

8.3 Use of user preferences for Houses In Reach website

The forms used to enter housing attributes generate a dynamic query for the user to display houses based on his or her preferences. The Clark and Clark [ClCl1989] and Hempel and Tucker [HeTu1979] housing attributes are entered using the form shown in

Figure 20 , Figure 21 and Figure 22 depicts the form used to input life values.

Intrinsic Attributes	
Cost:	
Price : From \$ <input type="text" value="500000"/> To \$ <input type="text" value="1000000"/>	
Size:	
Bedrooms : <input type="text" value="2"/>	Baths : <input type="text" value="2"/>
Units : <input type="text" value="2"/>	Floors : <input type="text" value="2"/>
Standard:	
Year build : <input type="text" value="2001"/>	House style : <input type="text" value="TownHouse"/>
Deck : <input type="text" value="Doesn't matter"/>	Backyard : <input type="text" value="Doesn't matter"/>
Notes:	
<input type="text"/>	
<input type="button" value="Save House Intrinsic Data Values"/>	

Figure 20. Intrinsic Attributes

Location (Distance)				
Work Address:				
Address: 77A Lowther Avenue	City: Toronto	Province: ON	Country: Canada	Zip Code: M5R 1C9
Recreation Address:				
Address: 23 Grange Rd	City: Toronto	Province: ON	Country: Canada	Zip Code: M5T 1C6
School Address:				
Address: 315 Bloor St W	City: Toronto	Province: ON	Country: Canada	Zip Code:
Close To DownTown: <input type="checkbox"/>				
Notes:				
<input type="text"/>				
<input type="button" value="Save House Locations"/>				

Figure 21. Location Attributes

Life Values

Excitement Family Freedom Happiness Health Inner Harmony Leisure Money Pleasure Security Togetherness	>> <<	Comfort
---	----------	---------

Save Life Values

Figure 22. Life Values

Each form has a button that enables the user to save the entered values. Submission of values from the intrinsic attributes form leads to the life values form. On submission of the life values form, the system directs the user to the house display page, discussed in Chapter 5. Based on the values input, the system selects the houses that is approximate to the user preferences and displays on the map.

The subsequent figures show display of houses for different life values but for the same user requirement. Figure 23 depicts the houses with life value “Comfort”. Figure 24 depicts the houses display for life value “Togetherness” and Figure 26 depicts the houses for life value “Family”.



Contact us
email : sgoyal@uvic.ca
Phone : 250-389-7737

Home	Member Sign-in	Articles	Market watch	Featured houses
Get email when prices drop	Key to buy your dream house in budget	Choose from our exclusive list	Target right price on right time	Keep an eye on the market

Please check to select houses --->

Bed:1
Bath:1
Price:\$139900
Toronto, Canada, ON.

House Listing For Third Preferences

Houses	Price \$
	\$598000 <input type="button" value="Select"/>

List of selected houses

34
36
40
14
33
9
8
35

Source Location Indicator

First House Preferences

Home	Member Sign-in	Articles	Market watch	Featured houses
------	----------------	----------	--------------	-----------------

Figure 23. Houses for Life Value “Comfort”



Contact us
email : sgoyal@uvic.ca
Phone : 250-389-7737

Home	Member Sign-in	Articles	Market watch	Featured houses
Get email when prices drop	Key to buy your dream house in budget	Choose from our exclusive list	Target right price on right time	Keep an eye on the market

Please check to select houses --->

Bed:1
Bath:1
Price:\$139900
Toronto, Canada, ON.

House Listing For Third Preferences

Houses	Price \$
	\$598000 <input type="button" value="Select"/>

List of selected houses

1
8
2

Source Location Indicator

First House Preferences

Home	Member Sign-in	Articles	Market watch	Featured houses
------	----------------	----------	--------------	-----------------

Figure 24. Houses for Life Value “Togetherness”

The screenshot displays the 'HOUSES in reach' website interface. At the top left is the logo, and at the top right is contact information: 'Contact us', email: 'sgoyal@uvic.ca', and Phone: '250-389-7737'. The main navigation bar includes: Home, Member Sign-in, Articles, Market watch, and Featured houses. Below this is a secondary bar with: 'Get email when prices drop', 'Key to buy your dream house in budget', 'Choose from our exclusive list', 'Target right price on right time', and 'Keep an eye on the market'. The central area features a map of Toronto with several house listings marked by red pins. A pop-up window for a selected house shows: 'Please check to select houses -->', a photo of a house, 'Bed:1', 'Bath:1', 'Price:\$139900', and a link to 'Toronto Canada, ON.'. To the right of the map is a sidebar titled 'House Listing For Third Preferences' with a table of 'Houses' and 'Price \$', a 'List of selected houses' (1, 8, 2), a 'Remove' button, and a legend for 'Source Location Indicator' and 'First House Preferences'. A 'Confirm' button is located at the bottom center of the map area. The bottom navigation bar repeats: Home, Member Sign-in, Articles, Market watch, and Featured houses.

Figure 25. Houses for Life Value “Family”

8.4 Summary

Life styles affect consumer preferences in choosing a home. Market analysis should not rely only on demographic data and census-based economic information, but should also incorporate information regarding consumer preferences. The incorporation of life values into real estate marketing tools will likely improve our understanding of individual decision-making in the real estate context. This in turn will lead to better options in the real estate market, which will lead to greater satisfaction in making decisions.

Chapter 9

Evaluation

In this chapter, we investigate the usability of the components we implemented for our Houses In Reach real estate website. First, we discuss the basic concepts of an analysis framework consisting of a set of cognitive dimensions. Next, we evaluate the parsing engine, the visualization engine, and the monitoring service and compare the usability of the Houses In Reach website contribute with that of the Zillow website.

9.1 Cognitive Dimensions

It is important that website users find the exact information they need. Any information that is not clear and that does not contribute its purpose promotes a site's disuse. Thus, the usability of the site is very important. We do not have the resources to conduct a meaningful and significant user study. Instead, we will investigate the usability of selected real estate websites using an analytical framework called cognitive dimension framework [Zill2007]. The aim of this framework is to provide an avenue to evaluate the usability of information-based artefacts. Artefacts are analyzed and a usability profile is derived that can assist in determining the suitability of artefacts for particular tasks.

The cognitive dimensions provide a fair assessment of how users will interact with the system; their use can help both designers and users to think and talk about the system. The cognitive dimension framework consists of 14 dimensions as listed in Table 13. Each dimension describes one aspect of a system, something that affects how a user will work with the system.

Table 13. Basic Cognitive Dimensions [GrBl1998]

Dimensions	Description
Abstraction	A class of entities, or a grouping of elements treated as one entity either to lower the viscosity or to make the notation more like the user's conceptual structure
Hidden Dependencies	A relationship between two components such that one of them is dependent on the other, but the dependency is not fully visible
Secondary Notation	Extra information in means other than formal syntax
Diffuseness	Verbosity of language
Premature Commitment	A constraint on the order of doing things that forces the user to make a decision before the proper information is available
Viscosity	Resistance to change
Visibility	Ability to view components easily
Closeness of Mapping	Closeness of representations to domain
Consistency	Similar semantics are presented in a similar syntactic style
Error-Proneness	Notation invites mistakes
Hard Mental Operations	High demand on cognitive resources
Progressive Evaluation	Work to date can be checked at any time
Provisionally	Degree of commitment to actions or marks
Role Expressiveness	The purpose of a component (or an action or a symbol) is readily inferred

We now discuss how these dimensions are used to discuss the usability of the “Houses In Reach” website. The goal of our evaluation is to assess a user’s ability to extract information compared to other websites. We used the questions listed in Table 14 to assess user tasks related to different website components based on our experience. Some of the results are outlined in Table 16 below where we compared the Houses In Reach website with the Zillow website for each cognitive dimension.

Table 14. Questions Regarding Cognitive Dimensions [BIGr2000]

Dimensions	Questions
Abstraction	<ul style="list-style-type: none"> • Does the system give us any way of defining new facilities or terms within the notation, so that we can extend it to describe new things or to express ideas more clearly or succinctly? What are they? • Does the system insist that we start by defining new terms before we can do anything else? What sort of things?
Hidden Dependencies	<ul style="list-style-type: none"> • If the structure of the product means some parts are closely related to other parts, and changes to one may affect the other, are those dependencies visible? What kinds of dependencies are hidden? • In what ways can it get worse when we are creating a particularly large description? • Do these dependencies stay the same, or do some actions freeze them? If so, what are they?
Secondary Notation	<ul style="list-style-type: none"> • Is it possible to make notes to oneself, or express information that is not really recognized as part of the notation? • If the notation was printed on a piece of paper that we could annotate or scribble on, what would we write or draw? • Do we ever add extra marks (or colours or format choices) to clarify, emphasise, or repeat?
Diffuseness	<ul style="list-style-type: none"> • Does the notation says what we want reasonably briefly, or is it long-winded? Why? • What sorts of things take more space to describe?
Premature Commitment	<ul style="list-style-type: none"> • When we are working with the notation, can we go about the job in any order we like, or does the system force us to think ahead and make certain decisions first? • If so, what decisions do we need to make in advance? What sort of problems can cause in our work?
Viscosity	<ul style="list-style-type: none"> • When we need to make changes to previous work, how easy is it to make the change? Why? • Are particular changes more difficult or especially difficult to make? Why?
Visibility	<ul style="list-style-type: none"> • How easy is it to see or find the various parts of the notation while it is created or changed? Why? • What kinds of things are more difficult to see or find? • If we need to compare or combine different parts, can we see them at the same time? If not, why not?
Closeness of Mapping	<ul style="list-style-type: none"> • How closely related is the notation to the result that we are describing? Why? (Note that in a sub-device, the result may be part of another notation, rather than the end product). • Which parts seem to be a particularly strange way of doing or describing something?

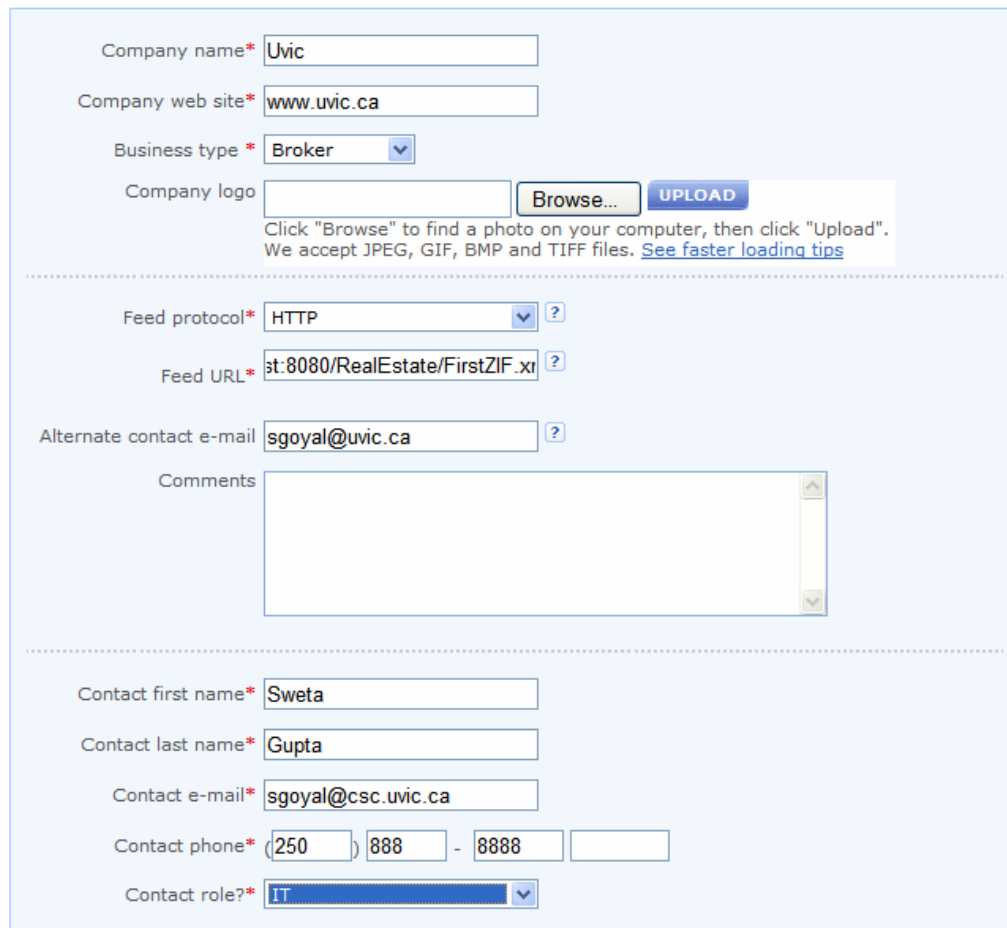
Table 15. Questions Regarding Cognitive Dimensions [BlGr2000] (Continued)

Dimensions	Questions
Consistency	<ul style="list-style-type: none"> • Where do different parts of the notation mean similar things? • Are there places where some things ought to be similar, but the notation makes them different? What are they?
Error-Proneness	<ul style="list-style-type: none"> • Do some kinds of mistake seem particularly common or easy to make? • Do we often find ourselves making small slips that irritate us or make us feel stupid? What are some examples?
Hard Mental Operations	<ul style="list-style-type: none"> • What kinds of things require mental effort with this notation? • Do some things seem especially complex or difficult to work out in our head ?
Progressive Evaluation	<ul style="list-style-type: none"> • How easy is it to stop in the middle of creating some notation, and check our work so far? Can we do this any time we like? If not, why not? • Can we find out how much progress we have made, or verify the stage of our work? • Can we try out partially-completed versions of the product?
Provisionally	<ul style="list-style-type: none"> • Is it possible to sketch things out when we are playing around with ideas, or when we are not sure which way to proceed? • What features of the notation help us to do this? • What sort of things can we do when we do not want to be too precise about the exact result we are trying to get?
Role Expressiveness	<ul style="list-style-type: none"> • When reading the notation, is it easy to tell what each part is for in the overall scheme? • Are there some parts that are particularly difficult to interpret? Are parts difficult to understand, but we put them in because it has always been that way?

9.2 Evaluation of Parsing Engine

The parsing engine of our real estate website registers RSS feeds and saves its contents to the database. Zillow also supports registering and saving of RSS feeds in ZIF or TIF format. Table 16 compares the parsing engines of Houses In Reach with Zillow using cognitive dimensions.

The outcomes of this comparison are available in Table 16 and are the result of a review that we conducted as a web administrator, who is registering feeds to the website. The form used in the evaluation of the parsing engine is depicted in Section 7.1. The web form for the Zillow comparison is depicted in Figure 26.



Company name* Uvic

Company web site* www.uvic.ca

Business type* Broker

Company logo

Click "Browse" to find a photo on your computer, then click "Upload". We accept JPEG, GIF, BMP and TIFF files. [See faster loading tips](#)

Feed protocol* HTTP

Feed URL* st:8080/RealEstate/FirstZIF.xml

Alternate contact e-mail sgoyal@uvic.ca

Comments

Contact first name* Sweta

Contact last name* Gupta

Contact e-mail* sgoyal@csc.uvic.ca

Contact phone* (250) 888 - 8888

Contact role* IT

Figure 26. Zillow Feed Registration Form

Table 16. Comparison of Parser Form and Zillow Feed Registration Form

Cognitive Dimensions	Parsing Engine	Zillow.com
Abstraction	The starting point is to register new RSS through the option of “Create new RSS” available on the web page. The rest of the available fields provide a notation like the user’s conceptual structure.	The start point is to specify the URL of the feed in ZIF or TEF format. The user has to specify name, company name, and URL of the company. This also supports the abstraction by providing the notation more like user’s conceptual structure.
Hidden Dependencies	The user registers new RSS. Once the registration is complete, the user can define various patterns for that RSS. If registration is not complete, then the button on the webpage to submit the pattern is disabled. With the concept of button enable and disable, the hidden dependency is seen clearly.	The registration form is a simple web form to register the feeds. Thus, no hidden dependency is available.
Secondary Notation	This is available through “Test RSS.” The use is to test the pattern defined for an item before saving it to the database.	Zillow developers do the validation of the feed submitted and they will inform the user in 5 to 8 business days.
Diffuseness	The text messages on the form are short and easy to understand by the user. The user has to type in or select the items from the list. All the commands are short and simple to understand.	The ZIF format is long to type. The predefined tags are available in the ZIF user manual [Zill2007].
Premature Commitment	The items to type in require care. For example “RSS String Function” requires an error-free java string function, otherwise the parsing engine will throw an exception.	Knowledge of the ZIF feed format is required before submitting the feed. This is a problem when the user is not aware of the language. The user needs to have a company website, impossible for FSBO users.
Viscosity	The patterns can be selected both before saving and after testing. However, a pattern cannot be changed once submitted.	Once submitted, the feed can be changed.
Visibility	All component parts are visible	The submission in the form is of a file. The other fields are visible and easy to understand.

Table 14. Comparison of Parser Form and Zillow Feed Registration Form (continued)

Cognitive Dimensions	Parsing Engine	Zillow.com
Closeness of Mapping	The parsing engine needs to determine the positions of the house items in a RSS. The parsing engine form is closely modeled for an actual RSS feed which minimizes new concepts to be learned.	The ZIF format tags represent the house items very clearly, but need to be learned beforehand.
Consistency	The form has the consistency of the text box or the selection box and the buttons to either to test RSS or to submit RSS for further parsing and saving it to the database.	The various form fields on a web page are consistent.
Error-Proneness	Typing errors may occur and the website needs careful entry of patterns.	No validation occurs of whether houses are actually available and fake houses can be registered.
Hard Mental Operations	The user has to go through the questions to enter the value for the pattern, which may be hard for a first-time user.	The elements in ZIF format are difficult to remember.
Progressive Evaluation	The user can test the registration of items before submitting it, but once saved it is not possible to view the saved patterns.	Once the feed is submitted, it is not possible to edit it and the user has no way to determine the progress of the registration within a week.
Provisionally	We can test to see if the parsing works, before saving it to the database.	None available.
Role Expressiveness	Some fields are not clear, like “yes” for RSS end operation.	The elements in ZIF formats represent its role. For example, the <StreetAddress/> element represents the street address very well.

Results

The registration is easy to do with the parsing engine component, but Zillow ZIF formatting requires learning and needs a web server to access. Moreover, the web administrator creates a new document every time new updates are available and uploads it onto the server to make it available for parsing at the Zillow end. However, the parsing engine is flexible for working with new updates once RSS feed is registered. Automatic tools can help update the RSS for the website owner. Nonetheless, updating the ZIF format is done manually by the site owner.

The Houses In Reach parsing engine's Test feature has better usability and it can be explained by our comments for the dimensions of progressive evaluation, provisionality, and secondary dimensions. The Test feature requires answers to the questions listed in Table 14 to enter the patterns for the form; the available sample help the user of this form to enter the values. Thus, this reduces the tough mental tasks. The parsing engine is prone to more errors, but the "Test RSS" feature resolves this issue. On the other hand, Zillow is less prone to data entry errors because the Zillow developers validate the feed before making it available to the public. Moreover, Zillow's ZIF format tags express the house items more clearly than does the parsing engine's registration form entry. This can be explained by our comments for the dimensions of Role Expressiveness.

In summary, the parsing engine registration form is more user-friendly than the Zillow format.

9.3 Evaluating the Visualization Engine

The visualization engine provides the mapping of house locations to Google map locations for display purposes as described in Chapter 5. Zillow also supports the display of houses on a map and supports the selection of houses for making future decisions. Table 17 outlines the evaluation of the usability of the visualization engine using our cognitive dimensions and compares it with that of the Zillow website. The outcomes as shown in Table 17 are the result of our investigation from a buyer's perspective. The

form used in the evaluation is available in Figure 10. An instance of the Zillow visualization engine is depicted in Figure 27 below.

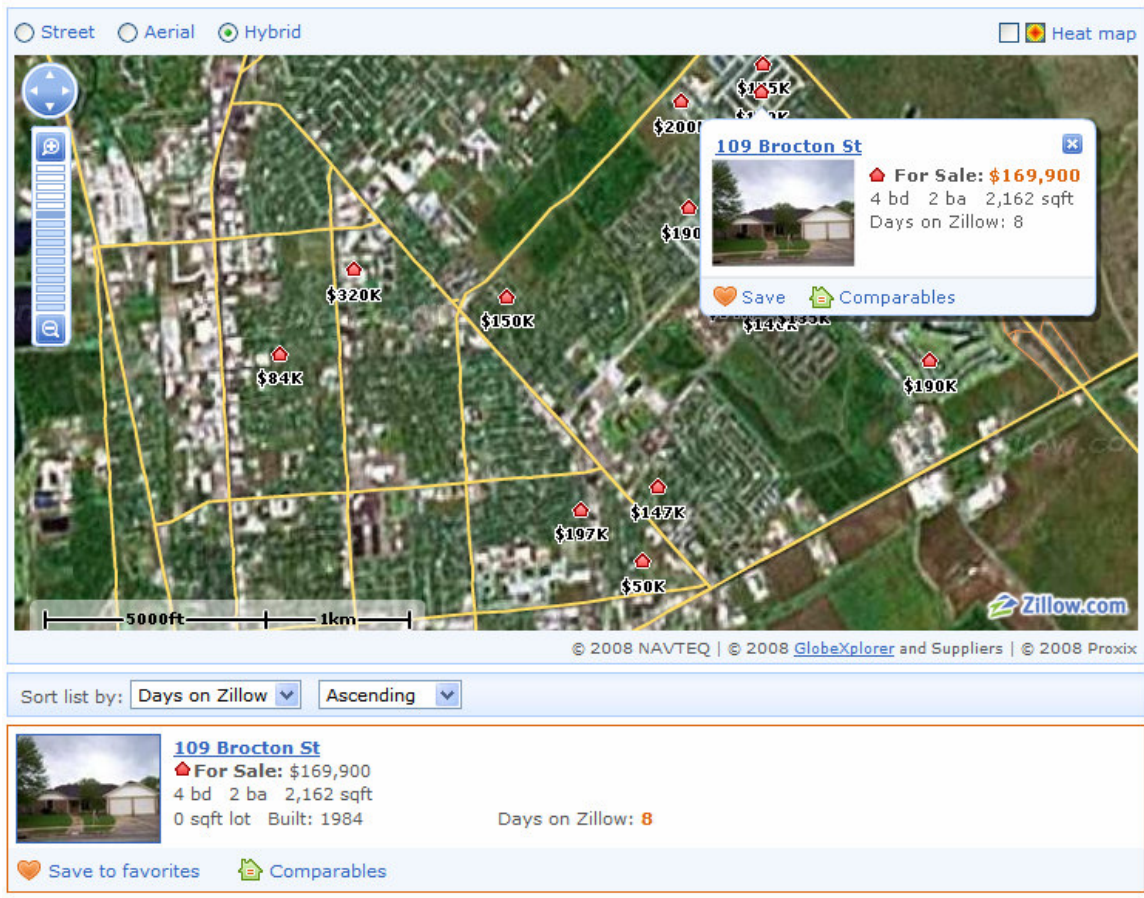


Figure 27. Zillow's Visualization Engine

Table 17. Comparison of Visualization Engine and Zillow's Mapping Service

Cognitive Dimensions	Visualization Engine	Zillow.com
Abstraction	The user can select houses from the brief information window where the abstraction of the information can be seen.	The user can select houses either from the list or from the map. The user can select houses as favourites.
Hidden Dependencies	None available	Houses are available on the map as well as on the list. The change in the map will affect the houses in the list.
Secondary Notation	A message indicates the houses already selected. The mouse over on the overlay will display the brief information of houses.	There is no secondary notation found for Zillow.
Diffuseness	The verbosity of language is not available	Houses listed at two places can be reduced to display at only one place.
Premature Commitment	The house locations should be entered before displaying the houses on the map.	At least one house location is required to display the houses on the map.
Viscosity	The selected houses once selected and saved can be updated only on next login.	The houses saved can be removed from the list if not required.
Visibility	The selection box is difficult to find the first time and the overlay's pointer does not point to the exact location.	The brief window for a particular house is difficult to identify for the house, as shown in Figure 27.
Closeness of Mapping	This is for locating and selecting the houses and satisfies its purpose.	This is for locating and selecting houses and satisfies its purpose.
Consistency	all artefacts available satisfies consistency	The save in map option in brief window and save favourites option in the listing mean the same, but are written in different ways.
Error-Proneness	The same house could be selected again.	There is less chance of error depicted
Hard Mental Operations	None available	None available
Progressive Evaluation	A user cannot specify different house locations after this page is displayed. As well, it is required to select houses once a user is on this page.	A house saved is at a different location and can be viewed by redirecting to a different page.
Provisionality	There is no artefacts available to help provisionality	There is no artefacts available to help provisionality
Role Expressiveness	It is difficult to distinguish first and second preferences.	This page displays houses well.

Results

The Zillow mapping service expresses its role clearly, and is less error-prone than our component. However, Zillow's house listing is available in two places, which does not make sense; this can be shown in an analysis of the diffuseness dimension. Zillow needs a more consistent information representation. In addition, our component has better visibility than the Zillow mapping service.

9.4 Evaluating the Monitoring Utility

Both, the Houses In Reach and the Zillow monitoring service send updates to potential buyers through email, but our monitoring service also provides updates on the webpage and a user can select and deselect the houses for future monitoring as discussed in Chapter 6. The user also has an option to add new houses from the newly suggested houses listed by the monitoring service. Table 18 depicts the evaluation of usability using cognitive dimensions and compares it with that of Zillow. The outcomes shown in Table 18 are the result of our investigation from a buyer's perspective. The form used in the evaluation is available in Figure 15. The updates sent by Zillow for a house are available via email as shown in Figure 28.



Figure 28. Zillow Monitoring Service

Table 18. Comparison of Monitoring Service with Zillow Monitoring Service

Cognitive Dimensions	Monitoring Service	Zillow.com
Abstraction	The check boxes and remove button is a good place to start. New houses listing can be updated from the list to monitor for the future.	No location defines new facilities or new ideas. A check box selects whether user wants a report on email or not. No option is available to define new terms.
Hidden Dependencies	None	None
Secondary Notation	No secondary notation is available and houses are updated and removed without warning.	None
Diffuseness	no language verbosity is found	no language verbosity is found
Premature Commitment	The user has to log in and should have houses selected for monitoring.	The user has to register by providing a valid email address and then can receive a report via email.
Viscosity	Once removed, a house cannot be reinstated.	No option exists to add or remove houses from the report. The user has to login to remove it manually from his or her account.
Visibility	The check box in the information window does not give an impression of selecting the houses.	The report is sent every month. The components are quite visible on the report.
Closeness of Mapping	The monitoring component performs the task and generates a report, which is close to the notation described.	The email report represents the updates of the houses.
Consistency	A selected house has a remove button and a new house entry has check box, making it inconsistent.	The notations in the report are consistent.
Error-Proneness	Houses can be removed by mistake without any warning.	None
Hard Mental Operations	None	None
Progressive Evaluation	ok	None available
Provisionality	The check boxes help to move the house from a new entry listing to the user's selected listing; it can move back.	None available
Role Expressiveness	ok	ok

The usability of our monitoring service could definitely be improved. It is inconsistent and some features could be improved. Zillow does not support all the dimensions either. It would be better for Zillow to have a web page where the user could update information similar to the web page for our monitoring service.

9.5 Summary

We have investigated the differences between a user's ability to extract information from our components and using the utilities available from Zillow. The two websites perform similar tasks, but they differ in accessing the information. Zillow performs services at the server side but our service is for the client side. Also Zillow does not restore the historic data as our Houses in Reach. Zillow is not flexible with respect to parsing feeds. The registration of the feeds process is more difficult to learn than ours. The Zillow monitoring service performs better than ours, but could be improved by providing a web page where users can update the information listed to match their requirements. Our visualization engine and Zillow's mapping service both performed well.

Chapter 10

Conclusions

This chapter summarizes our research findings, presents the main contributions of our research, and outlines future research.

10.1 Summary

Real estate is an information business and, consequently, deeply affected by information technology. One important phenomenon of the Internet world is “Really Simple Syndication” or RSS for short. RSS gives web users the ability to distribute content and make it widely available. Many websites use syndication feeds to add house listings to a website. However, even though RSS is easy to read and available in a standard format, different RSS feeds are available with different text formats, making it difficult for a web developer to consolidate all available feeds for a website. Accordingly, we have implemented an RSS parsing engine to parse any real estate RSS feed. This parsing component is useful for buyers, since it provides a consolidated listing of properties available on the web. It is also useful to the web developer, as it allows the web developer to test a real estate website with live data.

The web is also able to deliver specific information about the locations of particular houses and allows a homebuyer to browse through information concerning the houses. This helps homebuyers examine choices that match their requirements more efficiently than the conventional method of inspecting in person each potential property. Thus, the depiction of the location on the web can replace on-site visiting of a house; we have implemented depiction of the location using Google Map API as our visualization engine.

With the increase in the number of websites, competition among realtors has also increased. To survive in this competitive environment, realtors have to assist end users efficiently, at lower prices. By automating various processes within the enterprise, cost-effective websites can offer high quality services. Thus, the main goal of a real estate

website and their services must be to promote, facilitate, and enhance the buying and selling of real estate by making access to and use of real estate listings more user-friendly, timely, and effective. To achieve this goal, we must design an architecture that is flexible enough to upgrade to medium- or large-size applications. Thus, we have used the website model proposed by IMW for sharing distributed websites data and implemented it in the Houses In Reach website.

As the numbers of web users who search for houses online increases daily, it becomes increasingly important to develop a web-based product to cater to the various requirements of online users and help them buy the desired house efficiently. To do so, we must disentangle complex residential choices and preferences by proposing a data model and then implementing it on a website. For example, suppose a realtor is selling spacious homes. The challenge then becomes to identify homebuyers looking for spacious homes. In promoting these spacious homes to potential buyers, the realtor should focus on large households with a particular need for space. We have proposed and implemented a data model to achieve the goal of listing houses based on user preferences.

Buyers may also be provided with updates about their chosen houses. Sometimes, after considerable effort, they find that a house has sold or that the seller no longer wants to sell it. Hence, updates can save user's time. In addition, often listings are duplicated, as the seller lists with many agents in the hope of sealing a good deal as soon as possible. Thus, implementing a monitoring service to find new listings, to keep track of price trends, and to send alerts if a house is no longer available in the real estate market will help a buyer spend less time and avoid repeated logging-in to the website.

10.2 Contributions

The main contributions of this thesis include:

- Design and implementation of the Houses In Reach real estate engine, a prototype demonstrating that the goal of designing a flexible and extensible architecture for a personal real estate engine to draw data from different real estate sources can be achieved effectively.

- real estate model to accommodate different RSS real estate sources;
- user preferences model based on Lindberg *et al.* [LiGM1989] to display houses matching user preferences.
- customization model; and
- visualization model using Google Map;
- The implementation of a parsing engine to consolidate RSS feed information and provide a test bed of real time data for real estate engine web developers.
- An innovative architectural solution on how to add additional RSS feeds by end-users using a level of indirection.
- Design of a strategy for integrating different models to achieve data, control and presentation integration for a real estate mash up.
- Design of a strategy for integrating different technologies to implement the Houses In Reach prototype based on the IMW model.
- Design of a personal monitoring utility based on autonomic computing ideas (preliminary).
- Design of rudimentary support for identifying trends with respect to houses and areas (preliminary).

10.3 Future Work

We cannot claim that we have provided a complete solution for a real estate website; however, we are confident that our work is a step in the right direction. A number of other tasks could be automated:

- The current research for predicting user preferences and choices is limited to the Lindberg survey results. If we could model house buyer decision behaviour and perform an empirical study that would help predict decision behaviour of future buyers. This study would lead to a better understanding of consumer reasoning over the market choices. Another idea is to use data mining and machine learning algorithms to infer user preferences automatically.
- In addition, we could collect more interaction logs and organize more subjects to perform online experiments. This log would help develop a more accurate user decision behaviour model.

- We could compare the user preference data model with some other preference models, like that of migration intentions. [FHFZ2005]. The user preferences data model could be further explored using other methodologies like marketing, spatial cognition, psychology, and sociology to find more suitable premises upon which to design more lifelike algorithms.
- We could extend the parsing engine to parse and interpret the header links, parsing the detailed information, and saving it to the local database.
- The parsing engine for RSS could be further extended with a parsing engine for HTML or for ZIF. This would help to cover more house listings.
- Time consuming and resource-intensive user studies could be performed to evaluate and validate different aspects of the personal real-estate engine Houses In Reach.
- The limitations and difficulties for the RSS feeds could be improved by proposing a standard for housing information interchange.

Bibliography

- [AAA2004] Z. L. Anderson, G. D. Anderson and T.J. Anderson, "Web-Based Services for Real Estate: Model and Implementation," *Journal of Consumer Policy*, Vol. 6, No. 1, pp. 52-57, January 2004.
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/6294/28303/01265545.pdf?arnumber=1265545>
- [Mill2006] MillerSamuel, *Press Archives*, February 2006.
<http://www.millersamuel.com>
- [BJK2000] A. D. Bardhan, D. Jaffee and C. Kroll, "The Internet, E-Commerce and the Real Estate Industry," *Fisher Center for Real Estate & Urban Economics, Fisher Center Research Reports*, Report #6100, June 2000.
<http://repositories.cdlib.org/iber/fcreue/reports/6100>
- [Bett2007] B. J. Betts, "Zillow Frustrations," *Exceptional Marketing for Exceptional Homes News*, 2007.
http://www.beaubetts.com/news/archives /2007/07/zillow_frustrat.html
- [BIGr2000] A. F. Blackwell and T. R. Green, "A Cognitive Dimensions Questionnaire Optimized for Users," *Proceedings of Twelfth Annual Meeting of the Psychology of Programming Interest Group (PPIG-12)*, Vol. 1, No.2, pp.137-152, 2000.
<http://www.ppig.org/papers/12th-blackwell.pdf>
- [BSSB2000] M. T. Bond, M. J. Seiler, V. L. Seiler and B. Blake, "Uses of Websites for Effective Real Estate Marketing," *Journal of Real Estate Portfolio Management*, Vol. 6, No.2, pp.203-10, 2000.
http://www.cbi.msstate.edu/jrepe/volumes/Vol%203%20Issue%201/01.1_16.pdf
- [CIC11989] W.A Clark, and W. A. V Clark, *Human Migration*, Beverly Hills Press, 1989.
- [Cook2005] J. Cook, "Site Zooms in on Real Estate Listings," *John Cook's blog*, October 2005.
http://seattlepi.nwsourc.com/business/244104_homepages11.html
- [Cook2006] J. Cook, "Zillow Swamped, Crashes, Home-Valuation Website Attracts Millions of Visitors," *John Cook's Blog*, February 2006.
http://seattlepi.nwsourc.com/business/258825_zillow09.html

- [CrWi1999] K. Crowston and R. T. Wigand, "Real Estate War in Cyberspace: An Emerging Electronic Market," *International Journal of Electronic Markets*, Vol. 9, No. 1, pp.37-44, February 1999.
<http://crowston.syr.edu/real-estate/empaper.pdf>
- [FHFZ2005] C. Fredrickson, T. B. Heaton, G. V. Fuguitt and J. J. Zuiches, "Residential Preferences in A Model of Migration Intentions," *Journal of population and Environment*, Vol. 3, No. 3, pp. 280-297, March 2005.
<http://www.springerlink.com/content/j130h01245233145/>
- [GrBl1998] T. Green and A. Blackwell, "Cognitive Dimensions of Information Artefacts: A Tutorial," October 1998.
<http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDtutorial.pdf>
- [Grub2006] F. Gruber, "Comparing the Mapping Services," *Frank Gruber's blog*, April 2006.
<http://www.techcrunch.com/2006/04/17/comparing-the-mapping-services/>
- [Gunt2007] D. Gunter, "3 Real Estate Website Features Convert Visitors into Customers," *Internet Marketing*, April 2007.
<http://www.realtown.com/articles/marketing/3-real-estate-web-site-features-convert-visitors-into-customers>
- [HeCo2002] K. V. Henderson and L. B. Cowart, "Bucking E-Commerce Trends: A Content Analysis Comparing Commercial Real Estate Brokerage And Residential Real Estate Brokerage Websites," *Journal of Corporate Real Estate*, Vol. 4, No.4, pp. 375-385, May 2002.
<http://www.ingentaconnect.com/content/mcb/cre/2002/00000004/00000004/art00008;jsessionid=1m4r4h1ntnk3k.alice?format=print>.
- [HeTu1979] D. J. Hempel and L. R. Tucker Jr., "Citizen Preferences for Housing as Community Social Indicators," *Environment and Behaviour*, Vol. 11, No. 3, pp. 399-428, 1979.
<http://eab.sagepub.com/cgi/content/abstract/11/3/399>
- [InWS2007] Internet World Stats, 2007.
<http://www.internetworldstats.com/am/us.htm>
- [Java2007] Java.net, 2006.
<http://wiki.java.net/bin/view/Javawsxml/Rome>
- [Kern2007] A. Kernek, "REALTORS® – Get Ready for Zillow!" *Real Estate Internet Marketing*, 2007.
http://www.renewsletter.com/articles/article_zillow.htm

- [LiGM1988] E. Lindberg, T. Gärling and H. Montgomery, "People's Beliefs and Values as Determinants of Housing Preferences and Simulated Choices," *Scandinavian Housing and Planning Research*, Vol. 5, No.3, pp. 181-197, 1988.
<http://www.informaworld.com/smpp/content~content=a785737298~db=all>
- [LiGM1989] E. Lindberg, T. Gärling and H. Montgomery, "Belief-Value Structures as Determinants of Consumer Behaviour: A Study of Housing Preferences and Choices," *Journal of Consumer Policy*, Vol. 12, No. 2, pp. 119-137, June 1989.
<http://www.springerlink.com/content/0882uu638752xg11/>
- [Matt2007] Matt, "Zillow Vs. Homegain—Who Has the Most Traffic?" *Relevant blog*, June 2007.
<http://realiventblog.com/2007/06/15/zillow-vs-homegain-how-has-the-most-traffic/>
- [Mey2007] D. Meyers, "Don't Have an XML Feed Set Up? Where to Go to Get Started," *Geek Estate blog*, November 2007.
<http://www.geekestateblog.com/dont-have-a-xml-feed-set-up-where-to-go-to-get-started/>
- [Solu1999] Solution Center, "Pittsburgh Area Real Estate Professionals Form Preferred Alliance with REALTOR.com®. 1999," *Press Release*, November 2007.
http://resource.realtor.com/about/PressInfo.aspx?Page=1999/1999_0521_00
- [Scho2007] E. Schonfeld, "CyberHomes Emerges from Beta, With Claims of Better Real-Estate Data," *Schonfeld's blog*, November 2007.
<http://www.techcrunch.com/2007/11/11/cyberhomes-emerges-from-beta-with-claims-of-better-real-estate-data/>
- [Seth2003] M. Seth, "Proposing J2EE Based Architecture for Small/Medium Business Applications," 2003.
<http://www.developer.com/design/article.php/2175581>
- [Inte2007] Internet World Stats, "Internet Usage and Broadband Usage Report," 2007.
<http://www.internetworldstats.com/am/us.htm>
- [Gaw2006] GAW Associates, Inc., "The Top Realtors on the Net," *The Internet Real Estate News Letter*, September 2006.
<http://www.gawassoc.com/news/topreal.htm>

- [Wiki2007] Wikipedia, “Mashup Web Application Hybrid,” 2007.
[http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
- [WiFG2003] H. Wilson, S. F. Fabozzi and J. Gordon, “Why Real Estate? An Expanding Role for Institutional Investors,” *Journal of Portfolio Management*, Special Real Estate Issue, 2003.
- [Zill2007] Zillow, “Zillow® Listings Feed Guide: Technical Spec for Version 1.04 Zillow Interchange Format,” 2007.
http://www.zillow.com/static/pdf/feeds/ZillowListingsFeedGuidev104_COLOR_Final.pdf

Appendix A Glossary

- AJAX** *Asynchronous JavaScript and XML.* A scripting technique for silently loading new data from the server.
- API** *Application Programming Interface.* The specification of how a programmer writing an application accesses the behaviour and state of classes and objects.
- FSBO** *For-Sale-By-Owner.* A home that is offered for sale without the use of a real estate agent
- HTML** *Hyper Text Markup Language.* A markup language designed for the creation of web pages with hypertext and other information to be displayed in a web browser.
- IMW** *Internet Media Work.* Is a company specializing in web-based application implementation, database integration, and web development and hosting for businesses of all types.
- JSP** *Java Server Pages.* Extension of Java Servlet technology for combining Java server-side programs and HTML
- Mashup** A mashup is a web application that combines data from more than one web application.
- MLS** *Multiple Listing System.* A system that provides to its real estate members detailed information about properties for sale.
- MVC** *Model View Controller.* A design pattern that aims to modularize an application into three parts. The model represents the data for the application; the view represents the presentation; and the controller ties these two together and deals with user input.
- RSS** *Really Simple Syndication.* An XML-based system for aggregating and rapidly scanning information from blogs, news, current event Web sites, and other Web sites that update contents frequently.
- Servlet** A Java program that runs on a web server.

- Web Feed** A web feed is a data format used for providing users with frequently updated content.
- XML** *Extensible Markup Language*. A markup language that allows you to define the tags (markup) needed to identify the data and text in XML documents.
- Google** An URL to register for Map API : http://www.google.com/maps/api_signup
- Yahoo** An URL to register for Map API : <http://developer.yahoo.com/maps/>
- Zillow** An URL for registering feeds to Zillow: www.zillowfeeds.com
- ZIF** Zillow Interchange Format

Appendix B Set Up Instructions

The following steps help a novice user to set up Houses In Reach real estate web site for his or her personal transition:

- 1) Copy all files as they are in Installation directory
- 2) Create a folder JDK1.5
- 3) Create a folder JRE1.5
- 4) Install jdk-1_5_0_07-windows-i586-p.exe in the above directory
 - set up class path
 - right click on my computers in windows explorer
 - Inside Environment variables
 - JAVA_HOME = c:\java\jdk1.5
 - PATH = ;%JAVA_HOME%;%JAVA_HOME%\bin
- 5) Create a folder WTP
- 6) install wtp-all-in-one-sdk-R-1.5.2-200610261841-win32.exe
 - Install server
 - Unzip tomcat jakarta-tomcat-5.0.19.zip in c:\java
 - Eclipse window --> Preferences -->Server-->Installed Runtimes
 - Tomcat installation directory = C:\java\jakarta-tomcat-5.0.19
 - check Installed JRE
- 7) Eclipse Window-->Other--> Server-->Servers
 - Servers console--> New Server-->Tomcat v5.0 Server--> Finish
 - Test Server by starting it
- 8) Create new project --> Web-->Dynamic Web Project
 - Enter project Name

9) Install MYSQL server

- The instructions are available on the following site:

<http://www.wikihow.com/Install-the-MYSQL-Database-Server-on-Your-Windows-PC>

10) Unzip the file in the project created

11) Start tomcat which we are assuming set at port 8090 and project name is
RealEstateNewVersion

12) Start Houses In Reach application on browser using following URL

<http://localhost:8090/RealEstateNewVersion/Frame/HomePageWithuser.jsp>

Appendix C System Design

This section depicts the system design for major transition for Houses In Reach website.

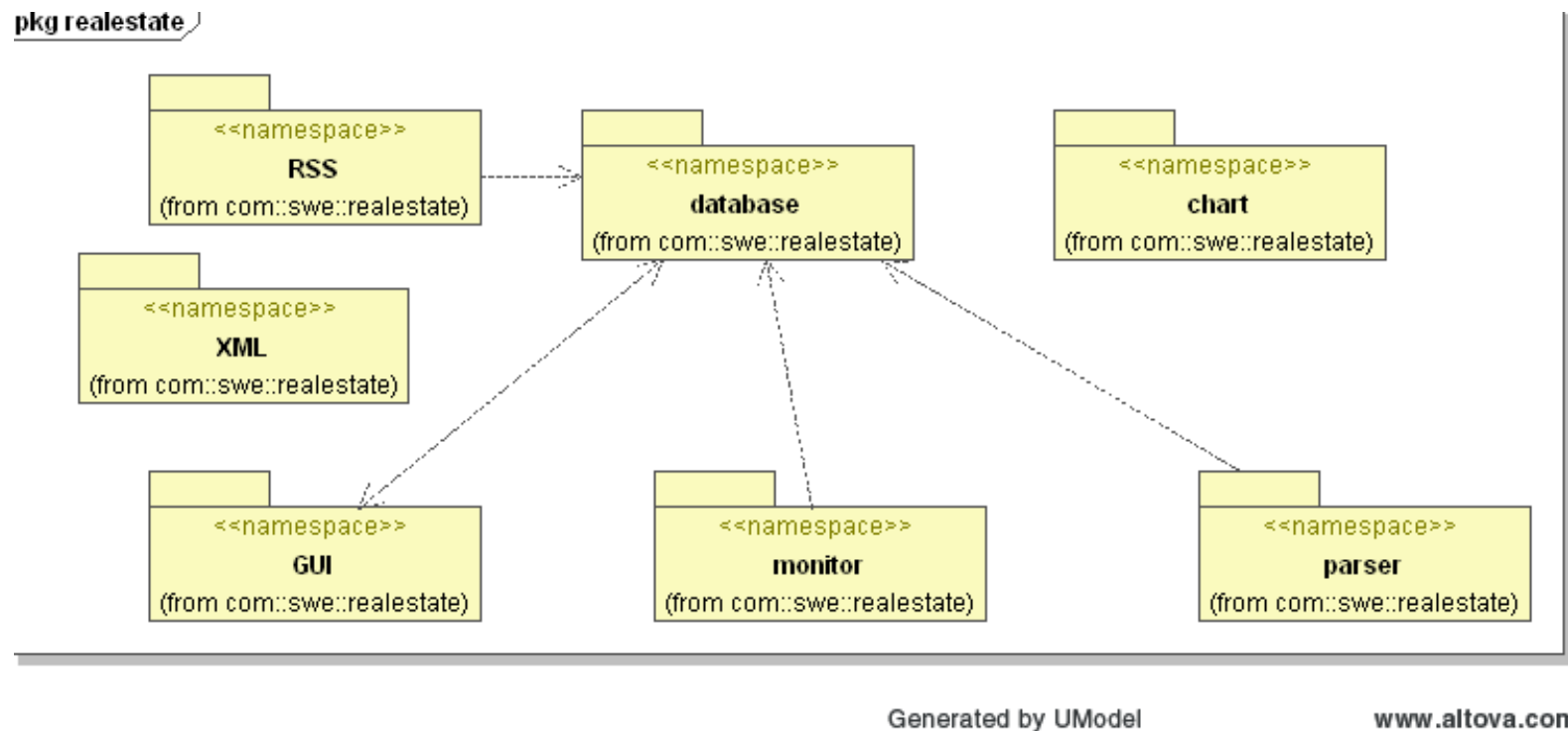
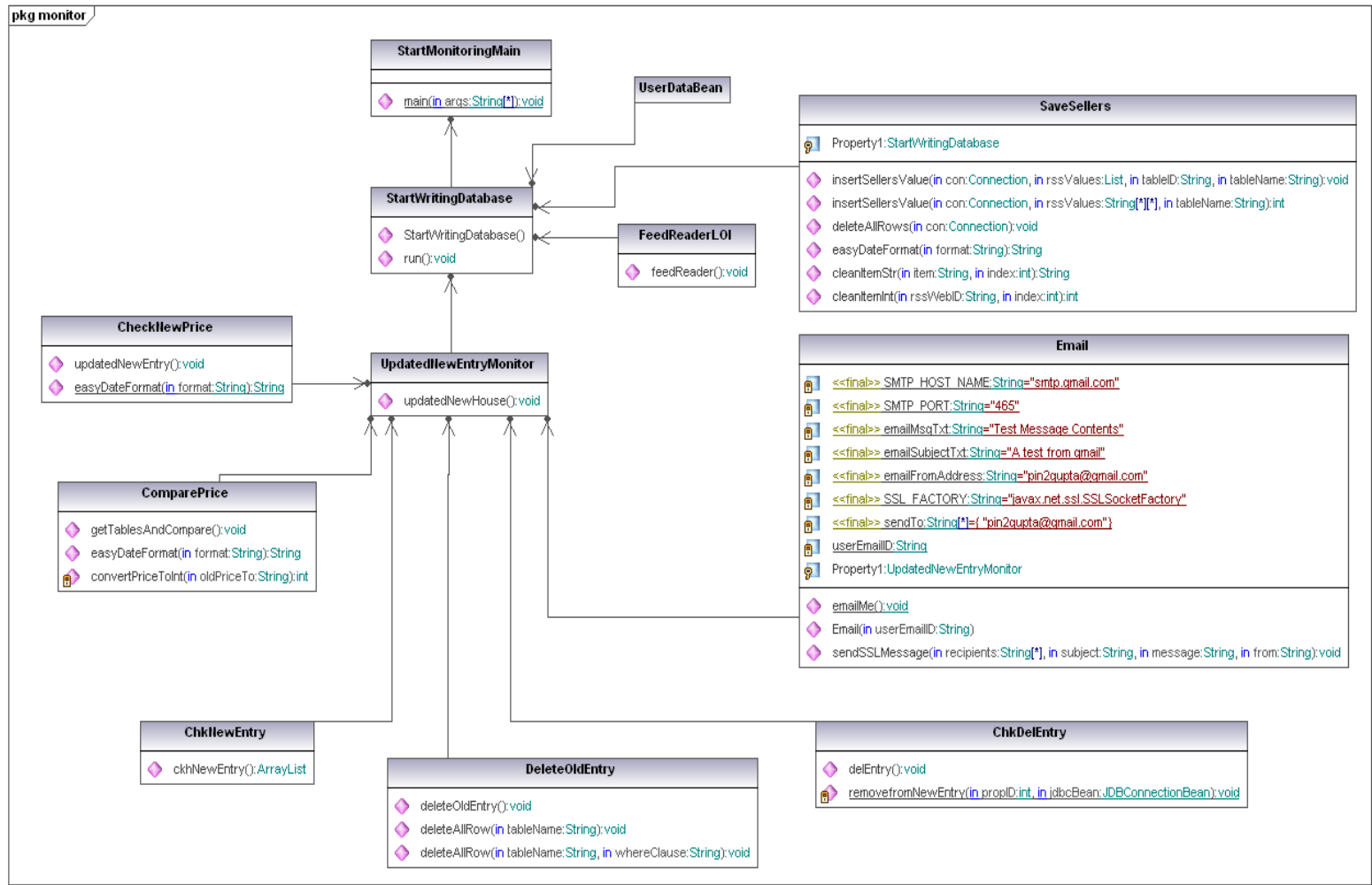


Figure 29. Package Diagram for Houses In Reach Prototype



Generated by UModel

www.altova.com

Figure 30. Class Diagram for Monitoring Services

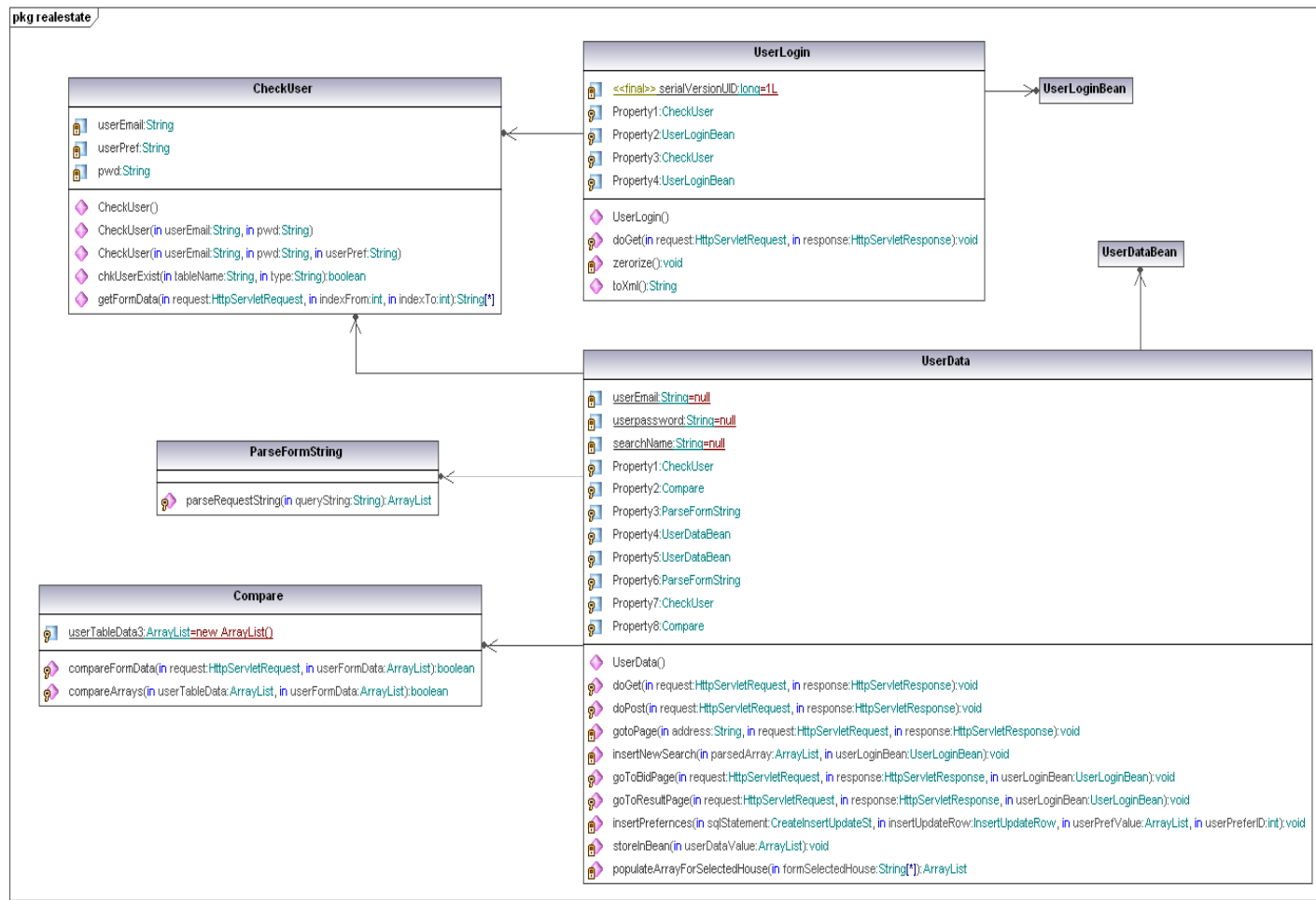


Figure 31. Class Diagram for User Interface

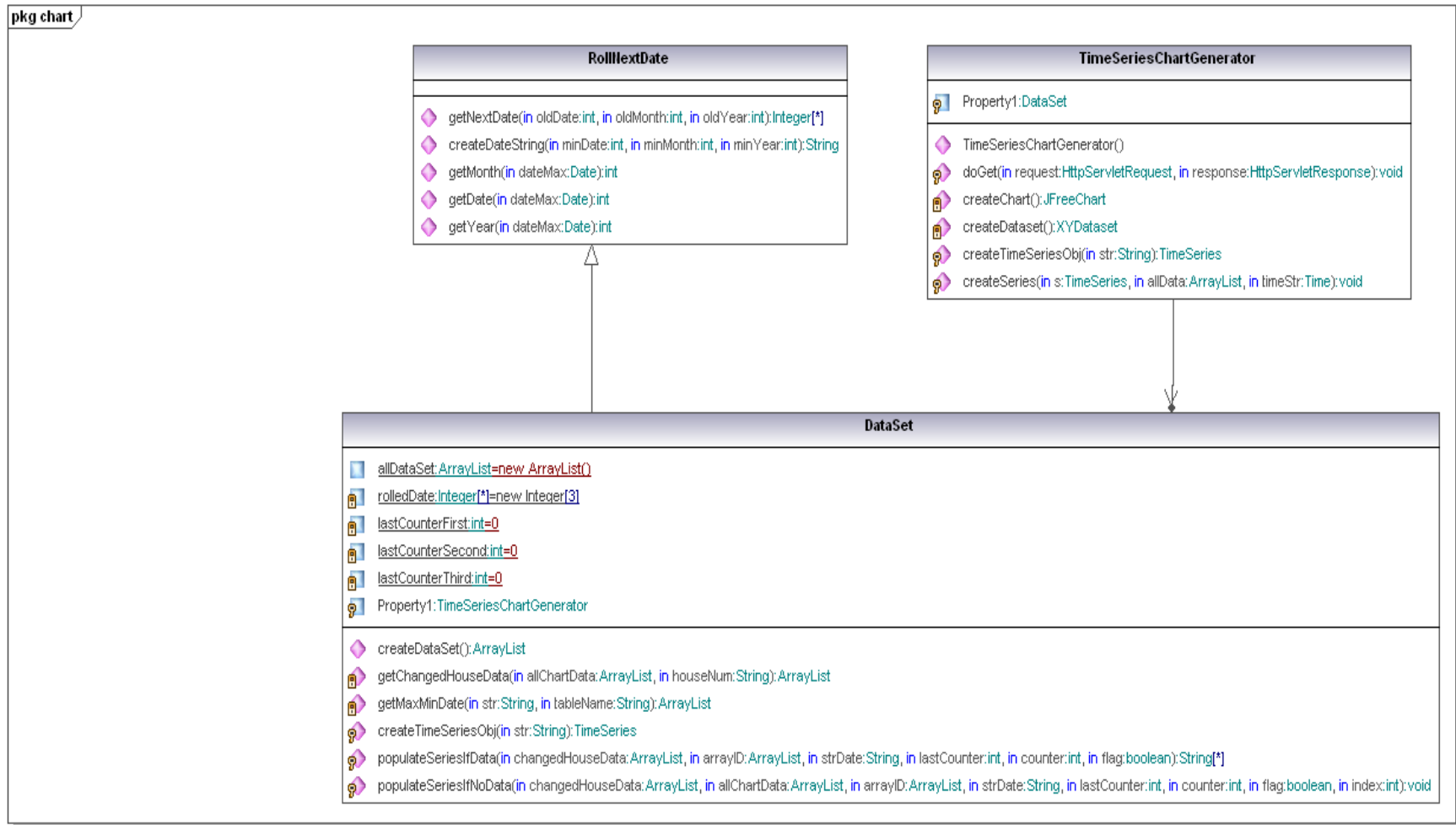


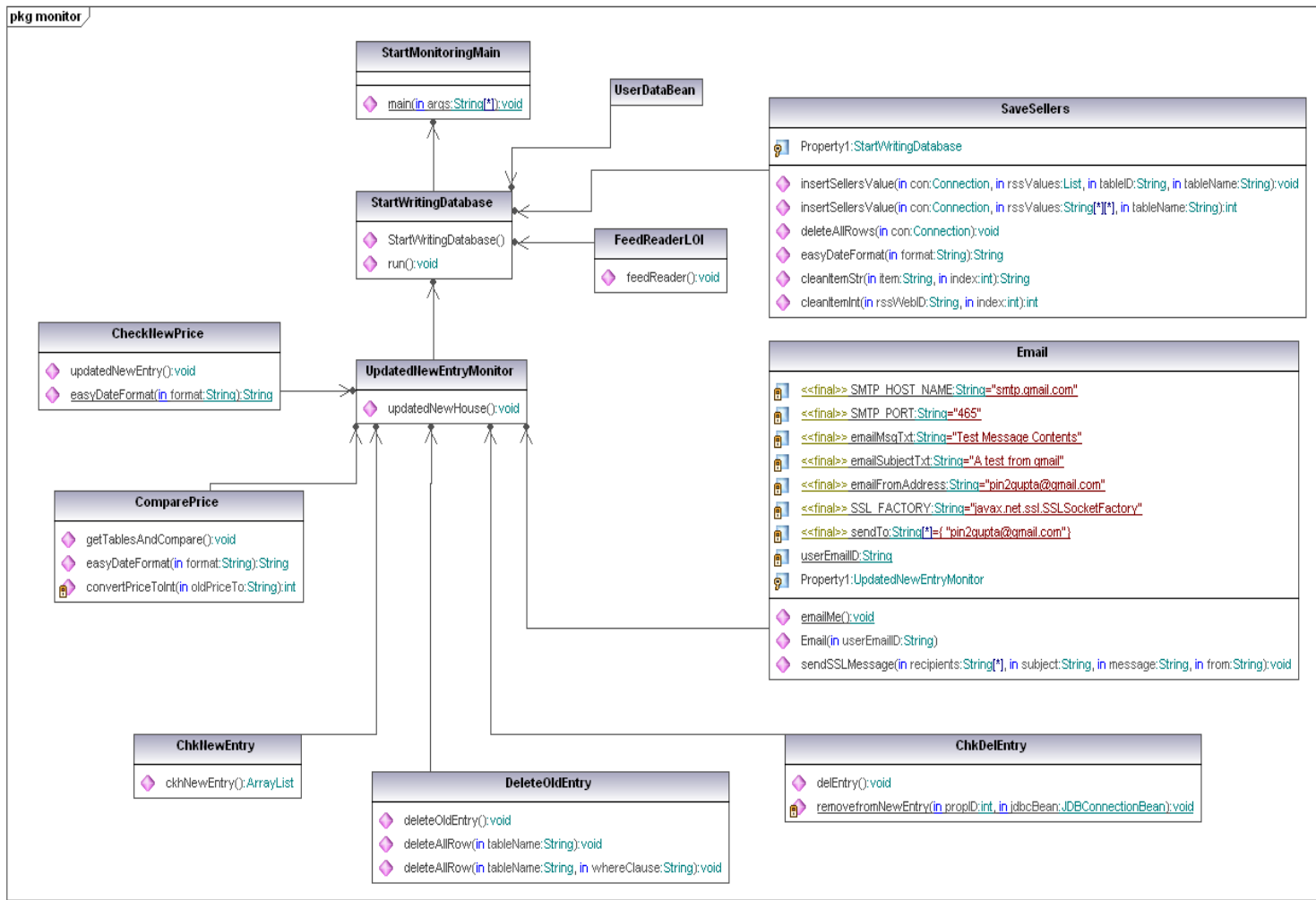
Figure 32. Class Diagram for Generating Chart



Figure 33. Class Diagram for setting User Preferences



Figure 34. Class Diagram for Parsing RSS feed



Generated by UModel

www.altova.com

Figure 35. Class Diagram for Checking updates

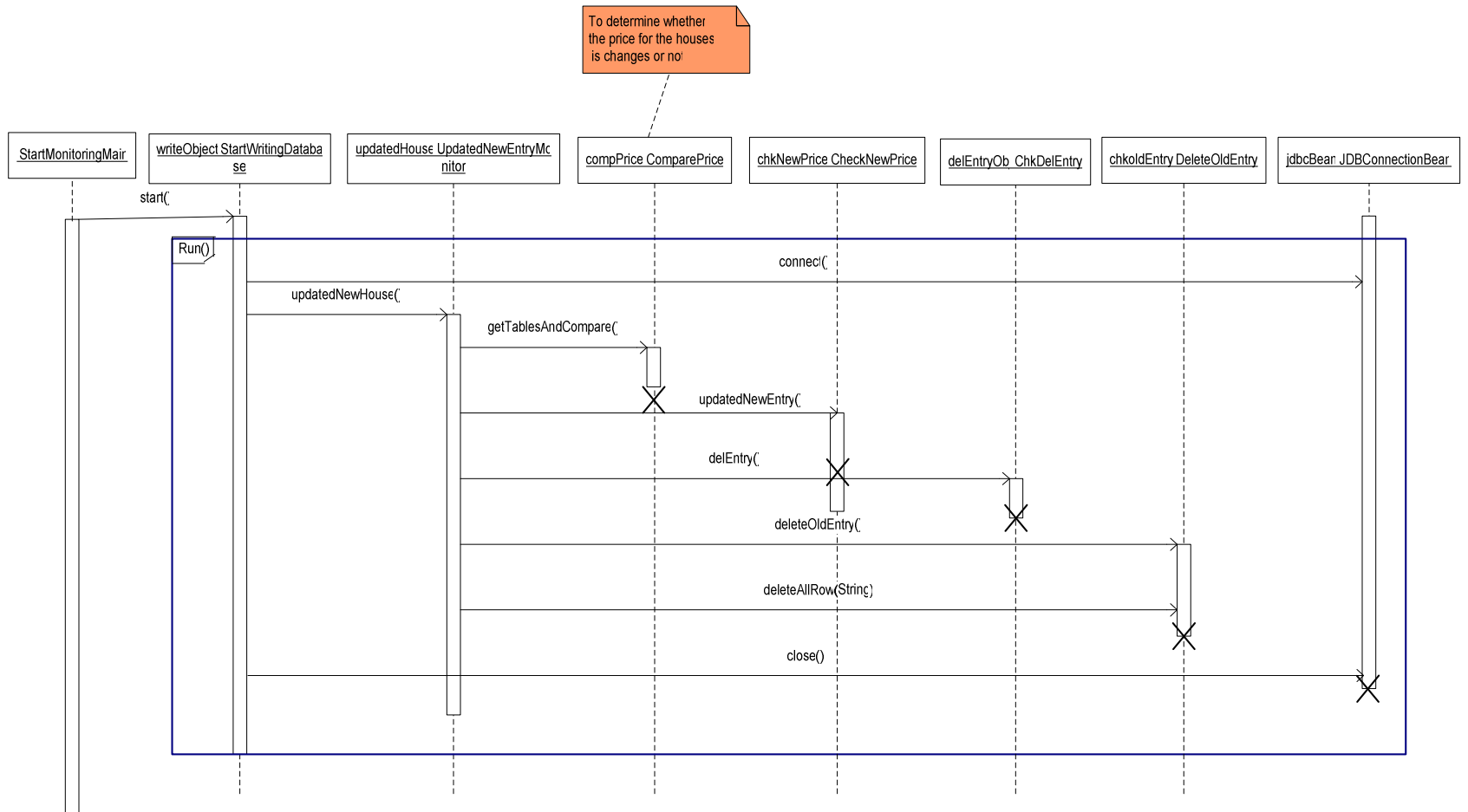


Figure 36. Sequence Diagram for Checking Updates

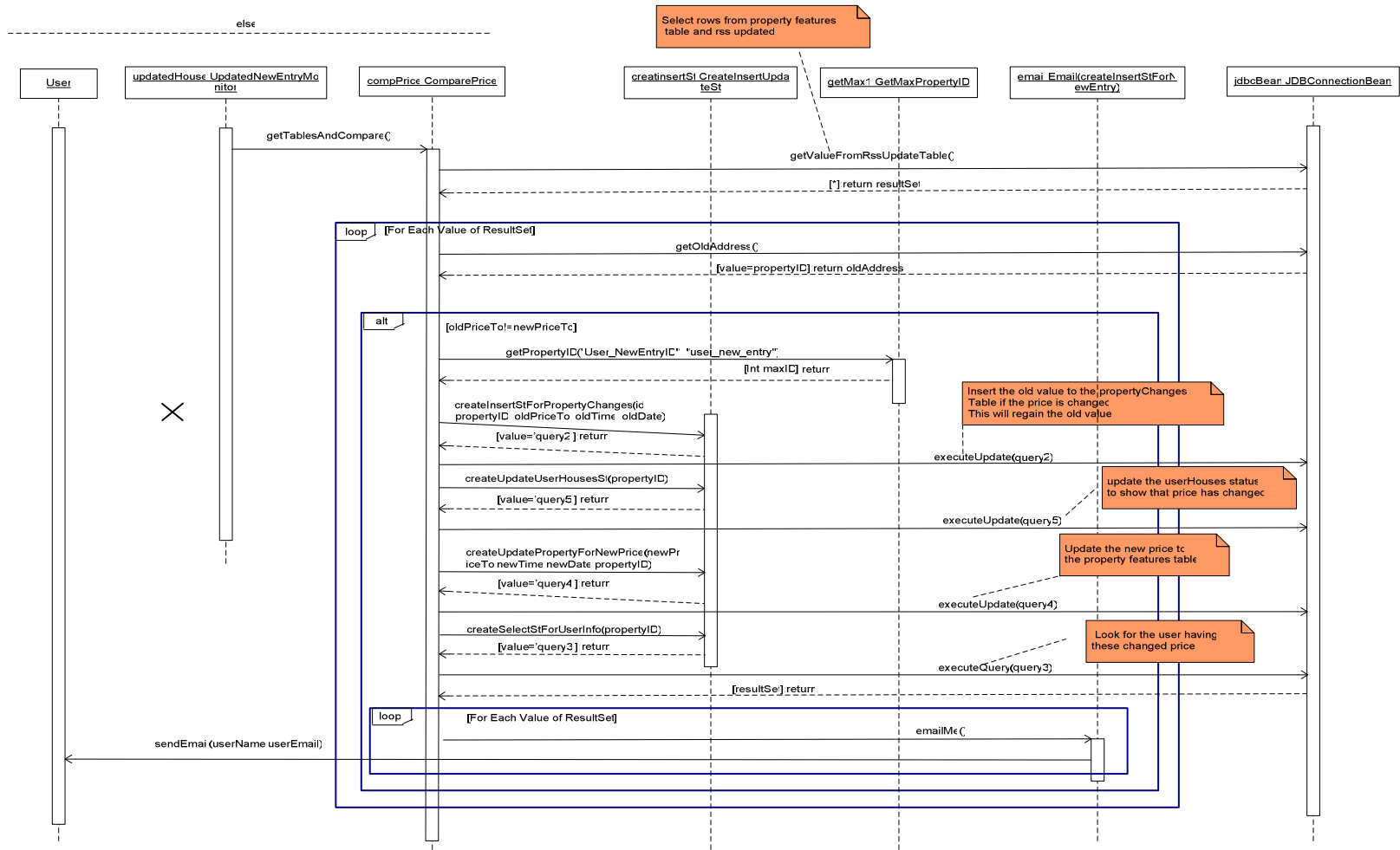


Figure 37. Sequence Diagram for Comparing Price

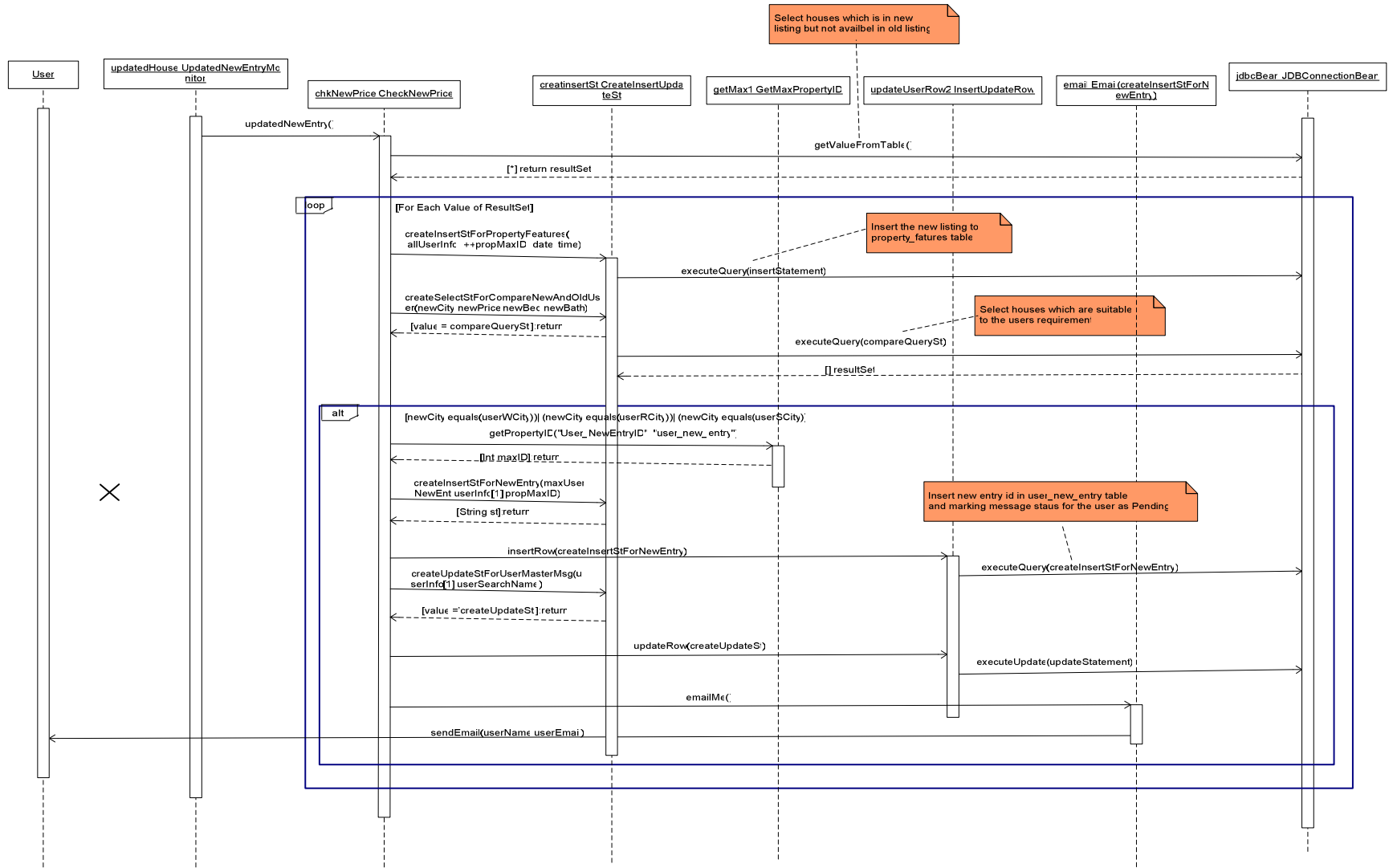


Figure 38. Sequence Diagram for Checking New Price

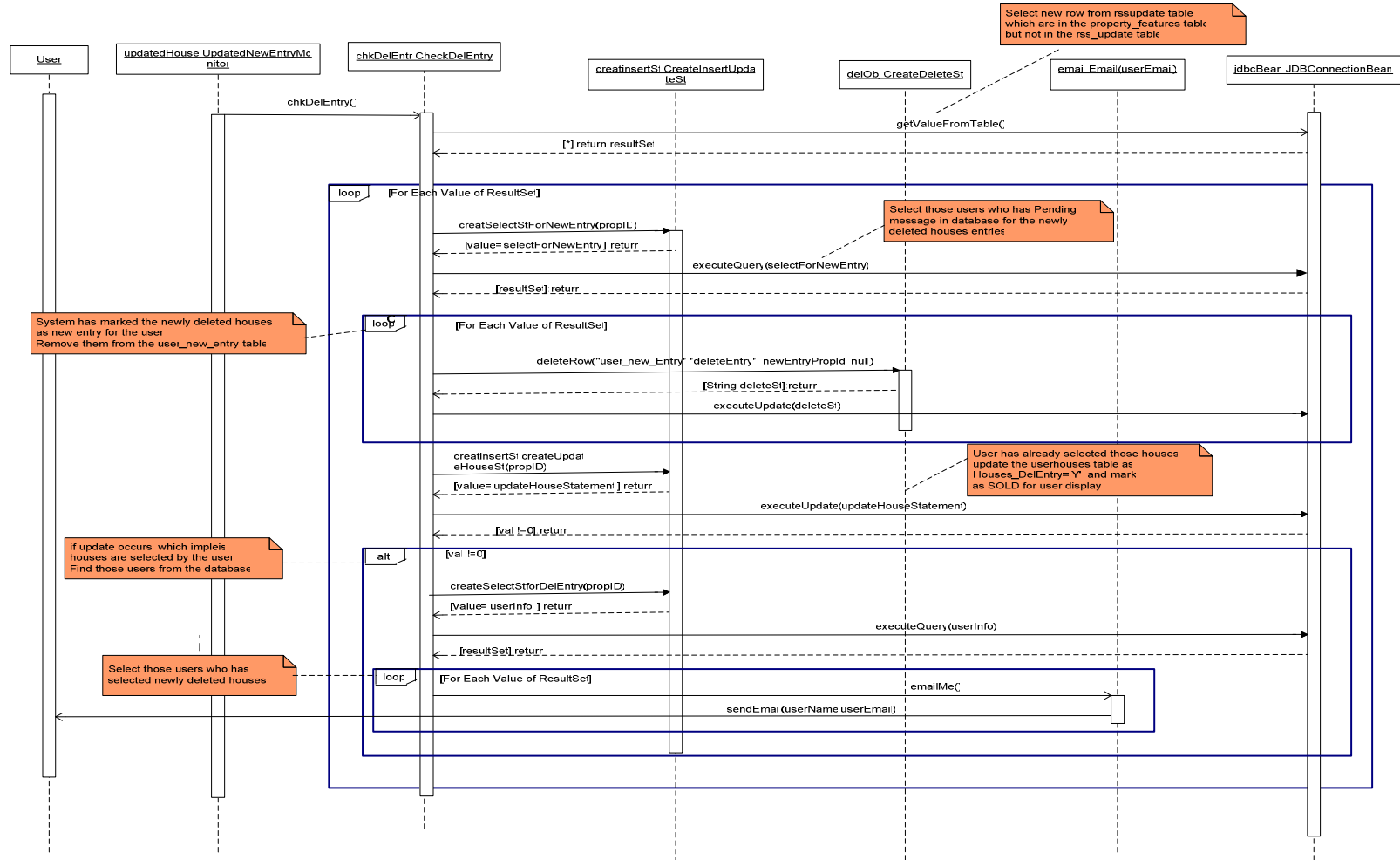


Figure 39. Sequence Diagram for Checking Deleted Entries

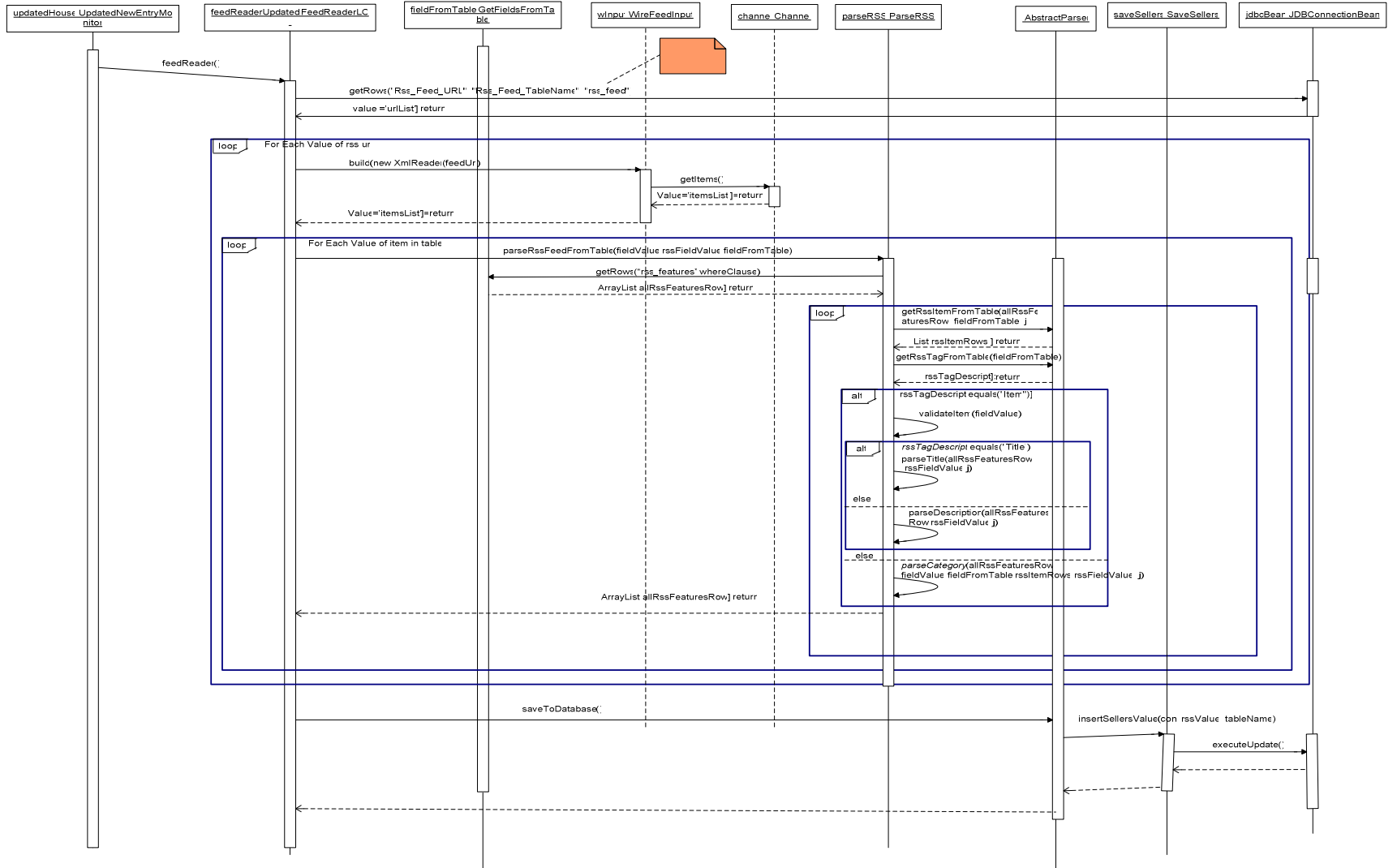


Figure 40. Sequence Diagram for Parsing RSS feed

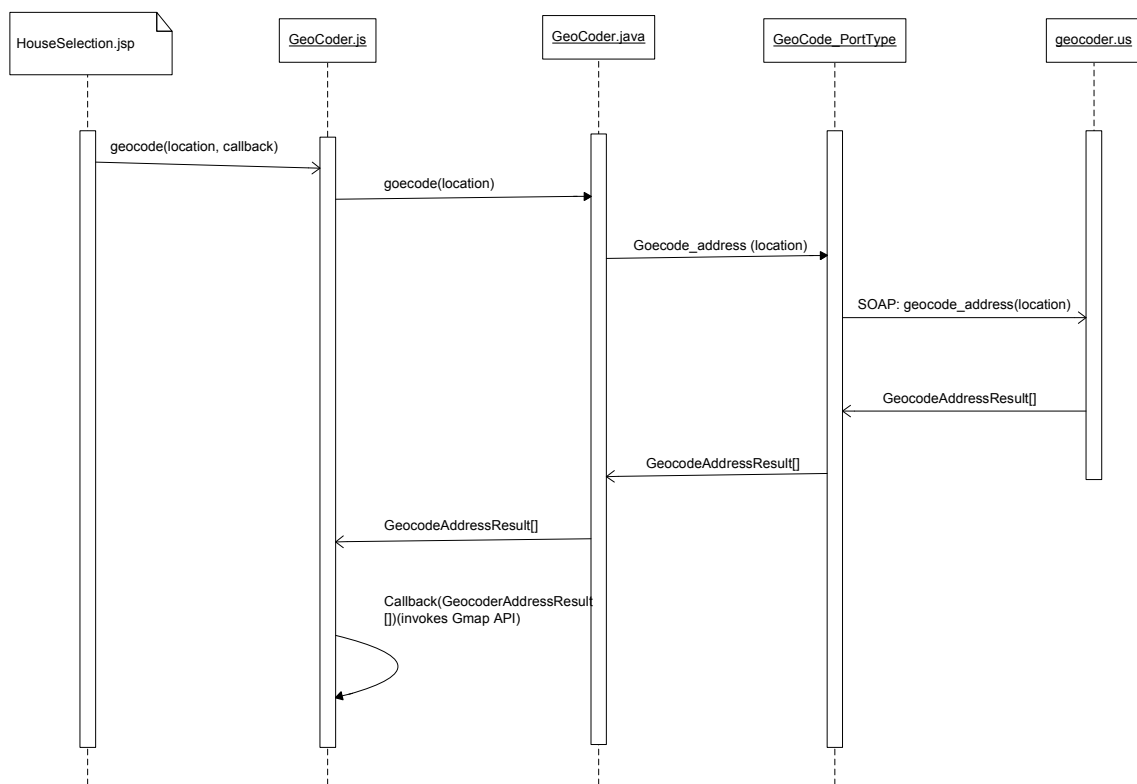


Figure 41. Google Map API

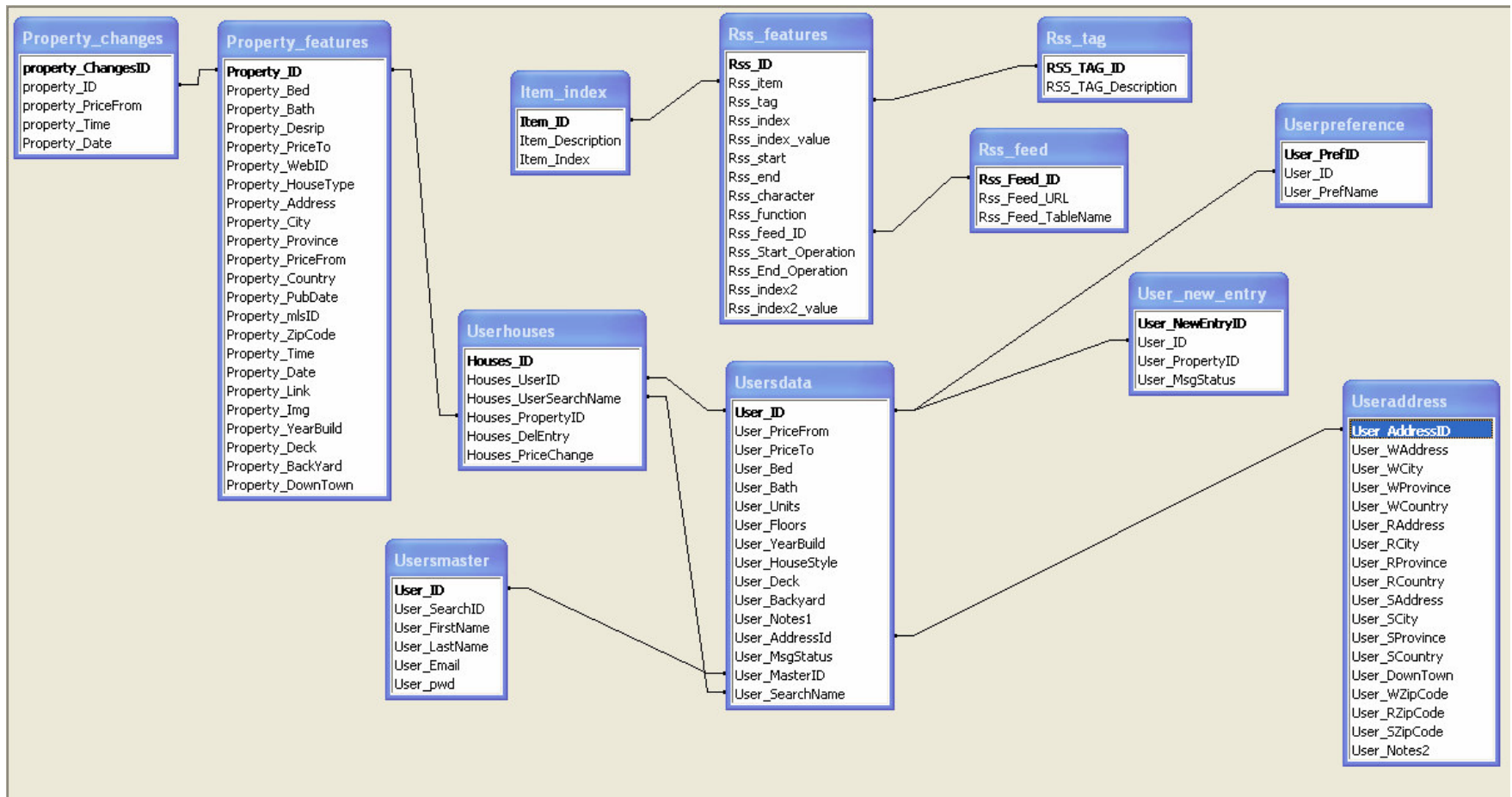


Figure 42. Database Design

Appendix D

Home Buyer Scenario

This section describes a scenario from a home buyer as follows:

Hi, I'm Neelam, a freelancer software developer. This is my story of buying a house.

We wanted to move to our second phase of life, that's a new house and a family. Since we were already on our way to having a family, we started thinking of a house as well. When we started thinking, we had few concerns, like the state of the market. In terms of housing prices, we discovered that this is a buyers market, which could give us more negotiating power. And that's the best time for buyers, when a lot of houses are available in the market and a wide range of choices as well as prices as per our need. So the first thing we asked was:

1. Is this the right time to invest in housing?

This was cleared and answered for us was "yes." We talked to our friends who own a house as well as found articles related to housing on the net. Once we were convinced that buying a house was a good idea, we moved further, to other issues.

2. What's our budget?

There are varieties of houses available for every budget, from condo to single house and others in between. After looking at our budget, we started looking for condos. Since there were condos available in the range of \$300k-\$450k, we had a good chance of finding a choice condo for a fair price.

3. The question then was what kind of condo we should be going for?

Two things came to our minds.

1. Location

2. Size

We had a wide variety to choose from. There were condos available in the city, but privacy was lacking, most were a little far from city but had good prices; many were in 2-3 story high rises. We also looked at luxury condos, posh locations, nice community, good facility, some with personalized offers to be ready in the future, some on seasonal discounts, also on sale by owners (pre-owned). We thought we'd start by looking at pre-owned condos since they might be more reasonable. So we started looking around in the city for "on-sale" boards, we searched availability on the internet, we visited realtor's sites, and we saw ads in papers and flyers, and started looking and talking to owners.

It was very time consuming and the availability was overwhelming for us.

It took a few months for us, and during this process we somehow felt that owning a condo is almost like living in an apartment, only it's ours. There were extra fees for certain communities, and the same apartment-type crowd living in them, a lot of condos was very close to each other, had no privacy, and had the same restrictions as in apartments. And on top of all that we never found any condo with such a price that we could just buy it. They always appeared costly for a condo. And we thought instead of buying a costly condo and feeling like living in a luxury apartment, we then we thought we should spend just a bit more and buy a house, where we could at least do whatever we want.

So we started looking for "for sale" signs for all kinds of houses, single, new, old, and town houses. Again, we collected a lot of flyers and every day we came back and short listed some of the flyers. It was again overwhelming, since there were tons of houses on the market. We had choices to go for a specific price range and location.

Since it was our first house, we were excited to see every house; we almost liked every house we saw and that confused us a lot. So we decided to think about what exactly we needed, like how many bedrooms and baths and all that. Once we decided, we short listed houses from our flyers and called them and visited them. We did it for months. Then we had a lot of choices based on location, prices, and our needs.

Now we were worried about the buying process and home loan process, since we had no idea and we were not familiar with all those big terms that realtors and banks say. We contacted our friends who recently bought a house and asked them a few things. But we were still not sure how the process exactly went without any errors. Since we read a lot of articles related to buying property on the net, we knew that there was a risk here, like fraud on realtor's side, For example, they might not do the legal work properly, or there could be fraud on the loan and it could be our headache later, or some hidden clauses in the paper we sign could get us in trouble. Some realtors don't explain the procedure well and they come up with issues later once you sign the contract with them. There are also problems in home ownership and so on. We thought it would be a good idea to go with a named reality business here, so we decided to have a good name reality person. Once we had our realtor, it was her job to search the houses as per our requirements and send us lists and update us whenever new listings and lower prices became available. We used to short list from her listing and started going with her to the houses to see them.

When we started seeing houses with her, we started thinking about of a lot of options like wood floors or carpet, big or small bedrooms, balconies and backyards, or 2-3 stories, or big kitchens, morning sun or evening sun, location, and many things like that.

So we demanded more specific listings from our realtor. After seeing lots of them we were still not there, we didn't think that we had found our dream house yet. Sometimes we thought of buying a costly house because those were the kind we liked. But very soon we came back to reality and started looking on our own on the internet in our budget range, since we had already increased our limit by going from a condo to houses

When we started searching for more houses in different locations, we found some good ones. So we decided to see them for ourselves, since it wasn't our realtor's territory.

She could only show us houses in the area she was supposed to. It's like they have different realtors for different areas. And now, after spending a lot of time with her, it didn't seem appropriate to go with another realtor. But if we were not satisfied with her listings, it was natural that we would not buy any unless we absolutely liked it.

So we started doing research on our own in different areas and came up with really good possibilities, or should I say as per our requirements in terms of location, prices, and size. We kept an eye on some we short listed with our realtor, as well as looked more. Again, we went to see houses on our own and short listed a few of them. Now we needed someone who could help us in this area.

About realtors, they charge you a commission so we looked at that too, talked to a couple of friends and found out that there are realtors who can give you some percentage of their commission back. We found one such realtor and we signed a contract with her. This time we wanted her only for legal issues and other formalities, not for searching because we already had short listed some of the houses.

Since it was buyers market, we waited for few houses to come down and also negotiated with the owners. And then we finally liked a house that seemed big at first time so we thought of a smaller one nearby, but then according to the market and size it was a bit overpriced, so we kept talking to them and negotiated and waited for the market to come down a bit. It was a kind of market in which houses were on sale for ever. Every time we visited this area we found all the sale boards intact. Nothing was gone--that gave us confidence in negotiating, since home owners don't want to keep a house for sale forever. Sometime they have already purchased another house so they need to sell the old one fast; sometime owners are stressed, due to leaving the country or for other reasons they want to sell a house fast. So we were looking for that opportunity as well.

We visited that big house we liked and thought maybe we'd increase our limit a bit and go for it since it has a 2 car garage. That's plus point in selling it and it could have been a good investment. So we talked to the realtors from the owner's side, negotiated

and then sent them contracts through our realtor a number of times. Some times we had an objection on something or sometimes the owners had some problem. So we went through it many times and finally reached on an agreement and we proposed our final contract. Meanwhile, we talked to few loan officers, tried to understand the terms and process of loans and short listed a few of them; every loan agent is different and we had to look for their commission, ease of talking to them, knowledge and patience, the lowest rates, honesty, reputation and experience, and his or her availability on time. Then we finally decided to go with one of them.

So now we had a final offer and our mortgage loan person. Now was the time to examine other things like insurance, closing costs, closing date, move in date, loan finalization date, down payment, title check, home check, direction of the house, other things. Once we did all that, the owners approved the contract, we all met on a particular date for closing the deal, and we got the house keys.

Now that big house we thought we would never buy was ours, and we were very happy that we did it. We think we got a good deal, but it was all our hard work and eye to detail and having patience that brought us here today and of course the market and internet helped a lot.

The market was important, because it was a buyer's market so we had . . .

1. negotiation power
2. a lot of houses to choose from
3. reasonable prices
4. time to make the decision. It wasn't a rush or a competition.

The internet was important, because . . .

1. listings were available in all the areas
- 2 the listings were according to specific needs
3. it provided a lot of information for first time buyers
4. it predicted the good time to buy

5. it provided tips on what to expect and how to negotiate and find a best deal
6. it provided updates on market availability
- 7 it provided almost everything a home buyer would want to know

Realtor issue

With the first realtor we used, we didn't sign any contract or commit anything, but finally we didn't go with her, as she was a bit suspicious that we are over crossing her or something. We cleared up her doubts that if we bought any house from her list, we'd buy it through her, but if we liked something which she didn't show us, then it didn't matter who we bought it from. She was little pissed off, because she had spent a lot of time as well as gas with us. But for us, it was a big decision to buy a house and big money and we couldn't have just gone for sake of her temper and buy something we didn't like. That was one thing which I didn't like, and it left me with a bad feeling, but we decided that she was a professional she would talk to everyone like, that so we finally overcame that issue.

Appendix E

Selected Houses In Reach Code

This section depicts the code used by parsing engine to parse the RSS feed and save it to the database. This section of code is referred in Chapter 7, Section 7.2

```

static void getRssValue(String rssTagDesc, List
allRssFeaturesRow, int j, String rss){
    int start =0;
    int end =0;
    int index=0;
    boolean IAmDone = false;
    if (!allRssFeaturesRow.get(j+5).equals("len"))
        start
        =Integer.parseInt((String)allRssFeaturesRow.get(j+5));
    else
        start = rssTagDesc.length();

    int noBedRoom =0;
    if(allRssFeaturesRow.get(j+3).equals("index")){
        if (allRssFeaturesRow.get(j+12).equals("index")){
            //if end depends on the another index
            String tableValue =
                (String)allRssFeaturesRow.get(j+13);
            end =
                Integer.parseInt((String)allRssFeaturesRow.get(j+
6));
            int indexValue = rssTagDesc.indexOf(tableValue);
            if ((tableValue.equals("FOR"))&&(indexValue==
-1)){
                //means index is SALE PENDING
                tableValue = "SALE";
                indexValue = rssTagDesc.indexOf(tableValue);
            }
            end = indexValue + end;
            IAmDone =true;
        }

        noBedRoom =
            rssTagDesc.indexOf((String)allRssFeaturesRow.get(j+4))
            ;
    }else if
    (allRssFeaturesRow.get(j+3).equals("startsWith")){
        if
        (rssTagDesc.startsWith((String)allRssFeaturesRow.get(j
+4))){
            if (allRssFeaturesRow.get(j+4).equals("$")){
                noBedRoom =rssTagDesc.indexOf("-");
            }
        }
    }
}

```

```

        end =noBedRoom;
        noBedRoom=0;
    }
    int hyphenIndex = rssTagDesc.indexOf("-");
    if
        (!allRssFeaturesRow.get(j+6).equals("len")){
            end = hyphenIndex;
        }
    }
} else if (allRssFeaturesRow.get(j+3).equals("US")){
    rssEachValue[arrayIndex]="US";
}

if (allRssFeaturesRow.get(j+6).equals("len"))
    end = rssTagDesc.length();
else if (allRssFeaturesRow.get(j+12).equals("NA")){
    index =
    Integer.parseInt((String)allRssFeaturesRow.get(j+
    6));
    end =
    getDescripLength(index,noBedRoom,descripValue);
}

if ((noBedRoom<0)|| (end<start)){
//no image is availble but has not chack with others
    rssEachValue[arrayIndex]="";
}
else {
//get operation, get start and end number to calculate
if (allRssFeaturesRow.get(j+10).equals("Add")){
    if (allRssFeaturesRow.get(j+11).equals("Add")){
        rssEachValue[arrayIndex]=rssTagDesc.substring(
        noBedRoom+start, noBedRoom+end).trim();
    }
    else if
    (allRssFeaturesRow.get(j+11).equals("NA")){

        rssEachValue[arrayIndex]=rssTagDesc.substring(noBedRoom+start, end).trim();
        if (rssItemIndex.equals("14")){ //image
            imageFlag = true;
        }
    }
    else if (allRssFeaturesRow.get(j+11).equals("ZE")){
        rssEachValue[arrayIndex]= rssTagDesc.substring(0,
end).trim();
    }
}
else if (allRssFeaturesRow.get(j+10).equals("NA")){
    if (allRssFeaturesRow.get(j+11).equals("Add")){//once
//twice

```

```

        rssEachValue[arrayIndex]=rssTagDesc.substring(start,
noBedRoom+end).trim();
    }
    else if(allRssFeaturesRow.get(j+11).equals("NA"))
        if (rssItemIndex.equals("14")){ //image
            rssEachValue[arrayIndex]=
                rssTagDesc.substring(0, end).trim();
            imageFlag = true;
        }
    else
        rssEachValue[arrayIndex]=
            rssTagDesc.substring(start, end).trim();
    }
    else if (allRssFeaturesRow.get(j+11).equals("ZE")){

        rssEachValue[arrayIndex]=rssTagDesc.substring(noB
edRoom+start, end).trim();
    }
}
}
}

```

Refer Chapter 7, Section 7.2

```

public void insertSellersValue(Connection con, List rssValues,
String tableID, String tableName){
try{

    GetMaxPropertyID getMax = new GetMaxPropertyID();
    int id = getMax.getPropertyID(tableID,tableName);
    System.out.println("ID:"+id);
    int i =0;
    try{
        Statement st = con.createStatement();
        for (Iterator iter =
            rssValues.iterator();iter.hasNext();){
            ArrayList rssValue =(ArrayList)iter.next();
            String bed =
                ((String)rssValue.get(i)).trim();
            if (bed=="")
                bed="0";
            String bath =
                ((String)rssValue.get(i+1)).trim();
            (bath=="")
            bath="0";
            String priceFrom =
                ((String)rssValue.get(i+2)).trim();
            String priceFrom1 =
                priceFrom.substring(1,priceFrom.length());
            String priceFrom2 =
                priceFrom1.replaceAll(",","");
            priceFrom = priceFrom2;

```

```

String priceTo =
    ((String)rssValue.get(i+3)).trim();
if (priceTo.equals(""))
    priceTo="0";
else {
    String priceTo1 =
        priceTo.substring(1,priceTo.length());
    String priceTo2 =
        priceTo1.replaceAll(",","");
    priceTo = priceTo2;
}

String area1 =
    ((String)rssValue.get(i+4)).trim();
String area2 =
    ((String)rssValue.get(i+5)).trim();
String area3 =
    ((String)rssValue.get(i+6)).trim();
String area4 =
    ((String)rssValue.get(i+7)).trim();
String area5 =
    ((String)rssValue.get(i+8)).trim();
String descript = (String)rssValue.get(i+9);
//replace the cahracter
String descript1 =
    descript.replaceAll("'", "");
String descript2
    =descript1.replaceAll("/","");
String descript3
    =descript1.replaceAll("\\","");
descript = descript3;
if (descript.equals(""))
    descript=null;
int webID=0;
if (rssValue.get(i+10)== "")
    webID=0;
else
    webID = Integer.parseInt(((String)
    rssValue.get(i+10)).trim());

String homeType = null;
if (rssValue.get(i+11)== "")
    homeType = "";
else
    homeType = (String)rssValue.get(i+11);
String mls =null;
if (rssValue.get(i+12)== "")
    mls = "";
else
    mls = (String)rssValue.get(i+12);

String publishedDate = null;

```

```

if (rssValue.get(i+13)==")
    publishedDate = "";
else
    publishedDate =
        (String)rssValue.get(i+13);
    //get current date and time to save it
    in features table
String date = easyDateFormat("yyyy-MM-dd");
String time = easyDateFormat("hh:mm");
String link = (String)rssValue.get(i+14);
String insertStatement =null;
int counter =0;
if (tableName.equals("rss_updated_feed"))
    insertStatement = "INSERT "+
tableName+" VALUES("+ ++id +", "+ ++counter +", '"+
bed+"', '"+bath+"', \"'+descript+\"\", '"+priceTo+"', '"+webID+"', '"+h
omeType+"', '"+
area1+"', '"+area2+"', '"+area3+"', '"+priceFrom+"', '"+area4+"', '"+p
ublishedDate+
"', '"+mls.trim()+"', '"+area5+"', '"+time+"', '"+date+"', '"+link+"')
";

    else if
        (tableName.equals("property_features"))
            insertStatement = "INSERT "+
tableName+" VALUES("+ ++id +", '"+
bed+"', '"+bath+"', \"'+descript+\"\", '"+priceTo+"', '"+webID+"', '"+h
omeType+"', '"+
area1+"', '"+area2+"', '"+area3+"', '"+priceFrom+"', '"+area4+"', '"+p
ublishedDate+
"', '"+mls.trim()+"', '"+area5+"', '"+time+"', '"+date+"', '"+link+"')
";

        int val = st.executeUpdate(insertStatement);
        i=i+14;
    }
    System.out.println("****Rows Inserted****");
}
catch (SQLException s){
    s.printStackTrace();
    System.out.println("Save Sellers : SQL statement
is not executed! "+s);
}
}
catch (ClassCastException e){
    e.printStackTrace();
    System.out.println("ClassCastException:"+
e.getMessage());
}
}

```

Appendix F

User Preference algorithm

```

String userPref=null;
String[] lifeValues = {"Comfort",
    "Excitement", "Family", "Freedom", "Happiness", "Health",
    "Inner Harmony", "Leisure", "Money", "Pleasure",
    "Security", "Togetherness"};
String[] houseAttributes ={"Recreation",
    "Schools", "Work", "Cost", "Size", "Standard", "Downtown"};

double[][] userPrefRatings= {
{4.08,4.19,3.03,4.25,1.92,3.11,2.11,5.28,2.00,2.03,1.33,3.06,3.03
},//Recreation
{3.42,1.89,3.94,3.92,1.36,1.08,2.19,1.89,2.17,1.11,2.44,2.36,2.31
},//Schools
{4.36,2.36,3.33,4.00,1.42,1.42,2.39,2.53,3.47,1.42,1.89,2.25,2.57
},//Work
{1.14,3.28,2.39,2.75,1.58,0.86,1.75,2.22,6.22,1.22,1.75,1.06,2.19
},//cost
{5.50,2.03,6.06,4.52,2.86,2.39,3.69,2.08,-
2.63,2.78,1.78,3.19,2.85},//size
{3.83,0.25,2.17,1.22,0.86,0.72,0.67,0.25,-
2.22,0.75,1.00,0.97,0.87},//standard
{3.36,2.17,1.00,2.44,0.53,-1.00,0.69,1.31,1.92,-
0.19,0.42,1.44,1.17},//downtown
};
UserDataBean userDataBean = new UserDataBean();
String[] userValue =
{"100", "100", "2", "3", "1", "2", "2005", "Condo", "Yes", "No",
    "169-3170 Tillicum Rd", "Victoria", "BC"};
    ArrayList fieldValue;
    boolean provinceFlag = false;
    boolean cityFlag = false;
    boolean countryFlag = false;
    boolean zipcodeFlag = false;
    boolean addressFlag = false;

//public static void main(String args[]){
public ArrayList getUserPrefFieldValue(String
userPrefItem,String prefType){
    //getLV();
    int IndexLV = getIndexLV(userPrefItem);
    String[] ValueHA = getRatings(IndexLV,prefType);
    fieldValue = getUserFieldValues(ValueHA);
    return fieldValue;
}

public String getLV(){
    return "Comfort";
}

```



```
return valueHA; }
```