Data modeling design and web interface

for

BTown Properties

Prepared by Group 13

Gaurav Ranade

Gouri Netravali

Jay Mercer

Ritesh Kavungal

Suresh Kanagala

Enrique Areyan

19 October 2011

Table of Contents

1. Introduction

1.1 Project Scope

BTown Properties is a web-based system where members and visitors can search the system's database for other members' properties. Only a member is allowed to contact another member to set meeting times to see a property. The system will allow members to upload information regarding their properties, maintain watch lists of properties and property features, and write reviews about other members' properties. Other features, such as social networking, may be added if time permits.

1.2 Purpose

In this document, we describe the conceptual features of *BTown Properties*, as well as its main design components. In particular, the document contains important definitions used throughout the project, the systems' functional requirements, and the data modeling and web interface design. The data modeling design contains both the ER modeling design and the design of the relational schema. The web interface design includes the design of the web pages and the transition among them. Finally, *BTown Properties*' business logic will illustrate the flow of data/control between the front-end and back-end, and the transition among web pages. (Note: we will use the male pronoun to refer to both male and female users)

1.3 Definitions

The following terms will be used throughout the document.

Property: physical unit with a unique address in the real world and whose owner has voluntarily agreed to upload its information into the system.

Users' accounts: there are two types of *users' accounts*:

1. Administrative account: a user with this type of account will have permission to access and perform operations on all the data in the system's database. This user can enable and disable other users' accounts. This user is simply called an *administrator*.
2. Member account: a user with this type of account can enjoy all the benefits of the systems. A user with a member account is simply called a *member*. A member cannot modify or erase data of another member.

Visitor: person that visits *BTown Properties* but does not have any type of user account. A visitor can only search the property's database but will need to sign up for a member account to enjoy the system's other benefits.

Watch list: collection of properties defined by a member according to his preferences. There are two types of watch lists:

1. Properties' watch list: a member can add a specific property to this list.
2. Features' watch list: a member can add specific features to be watched into this list, e.g. watch all properties with two bedrooms and 1.5 bathrooms; the watch list will contain all properties that match such features.

2. System Features (SF)

In this section, we present the functional requirements or major services provided by the system. The section is organized by detailing each major feature along with its priority. A priority is a natural number from 1 to 3, with 1 being a high priority feature (core functionality), 2 a medium priority, and 3 a low priority (nice-to-have feature).

2.1 Visitor Sign-Up. Priority 1.

2.1.1 Description: a visitor should be able to sign-up and become a member of *BTown Properties*. A member is uniquely identified by his primary email address, which is the main point of contact between the member and the system.

2.1.2 Functional Requirements: the user must input the following data. (*) means the data is obligatory (this same notation is used throughout the system features):

- First and Last Name (*)
- Primary Email Address (*)
- Date of Birth
- Nickname
- Password (*)
- Phone Number

Once the data is validated, the member will receive a link to his email address to activate his member account. Once the account is activated he can login into the system (SF 2.2).

2.2 Member Login/Logout. Priority 1.

2.2.1 Description: members can login into the system where they can access their data (SF 2.3), change their personal information, review their watch lists (SF 2.6), and review any contacts that might have been received or sent (SF 2.5). Also, members can logout of the system.

2.2.2 Functional Requirements: the member inputs his primary email address (SF 2.1) and password to login in the system. If the credentials are valid, a session will be created. If not, an error message will be displayed. Finally, a link will be provided for the member to logout. The logout function will destroy the session previously created.

2.3 Add, modify and delete properties. Priority 1.

2.3.1 Description: a member can add, modify and delete only his own properties. Once a property is in the database, it will be accessible to both users and visitors.

2.3.2 Functional Requirements: the member inputs the data of a property through a form with the following fields:

- Address (*), e.g., 980 W. Basswood St. Apt. C, Bloomington, IN, 47403.
- Description.
- Photos (0 to 5).
- Features, e.g., number of bedrooms, bathrooms; utilities are included or not, etc.
- Vicinity information, e.g., close to IU Campus (less than 3 miles).
- Rental/Sale price

The member may, at any time, modify or delete this information.

2.4 Property view. Priority 1.

2.4.1 Description: members and visitor can view the details of any single property.

2.4.2 Functional Requirements: an interface will show all the data of a single property in a centralized manner. The property view will display the property's information and any review that it may have. A user can access this view directly if they know the property's id in the system or they may reach it via the search function (SF 2.5). A Google Map will be shown pointing at the property's location.

2.5 Search and list properties. Priority 1.

2.5.1 Description: visitors and members can search the properties database to display a list of properties that match criteria selected from a list of property attributes. Example search attributes would include as min/max rent, number of bedrooms/bathrooms, pool, etc. A basic, "quick," search would result in a list of properties that match the given criteria. A more advanced "skyline query" search feature may be implemented to filter and rank the results so that the user can select the most promising ones.

2.5.2 Functional Requirements: the basic search would consist of a simple SQL SELECT query, whereas, the skyline query would be implemented using a nested SQL Query, or a Block Nested-Loop Algorithm within a stored procedure. After inputting the desired search criteria and submitting the form, the user would be presented with a list of properties matching selected criteria. Clicking on a search result would allow them to view the single property. (SF 2.4)

2.6 Contact a member. Priority 1.

2.6.1 Description: a contact function will stand for a "rent" function given that before the actual physical rent of a property takes place, the parties involved must undergo other processes in the real world and outside the scope of the system. Instead, a member can contact another member and set a meeting to see a property**.**

2.6.2 Functional Requirements: a member viewing the details of a single property (SF 2.4) may click a *Contact* link, which will display a form with fields:

- Property (fixed with the current property)
- Owner Member Name (fixed with the current owner)
- Contacting Member Name (fixed with the current member)
- Commentary
- Proposed Date of Visit

Once the member submits the form, a copy will be sent to the owner member via email.

2.7 Watch Lists. Priority 2.

2.7.1 Description: members may maintain two types of watch lists: properties' watch list and features' watch list.

2.7.2 Functional Requirements: the system will indicate to a member who is viewing (SF 2.4) a single property that is in either his properties' or features' watch list, that he is "watching" this property. Otherwise it will allow him to add the property only to the properties' watch list.

The features' watch list depends on specific features of a property and thus may not be added one at the time. For instance, a member may define a features' watch list to watch all properties with 2 bedrooms and a pool. In the future, if a property with such characteristics is uploaded into the system, an email will be sent to the member notifying him of the new property that might be of interest to him.

2.8 Review properties. Priority 2.

2.8.1 Description: a review is a commentary made by one member about a property. A member may review another member's properties but he may not review his own properties. A message will be display to members about *BTown Properties'* non-offensive language policy. Moreover, the system will check and disable any potential offensive language in a review. The reviews will be visible to both members and visitors in the view of a property (SF 2.4).

2.8.2 Functional Requirements: a member may click on a *Review* link inside the view of a single property (SF 2.4), and fill out a review form with the following fields:

- Commentary (*)
- Ranking (1-10) (*)

Once a commentary is submitted, it will go through a "bad word filter" that will check each word of the message and delete any potential bad word that are matched against a "bad words" repository. If a message surpasses a certain number of "bad" words, it will be placed under quarantine until an administrator reviews it.

2.9 Extra Features

The following features are not essential to the success of the project and thus, will be implemented only if time permits.

2.9.1 Share properties. Priority 3.

Description: visitors and members may share a property's information through Facebook

Functional Requirements: in (SF 2.4) a "Share" link will allow members and visitors to automatically upload a property's information in their Facebook wall.

2.9.2 *BTown Properties* mobile version. Priority 3.

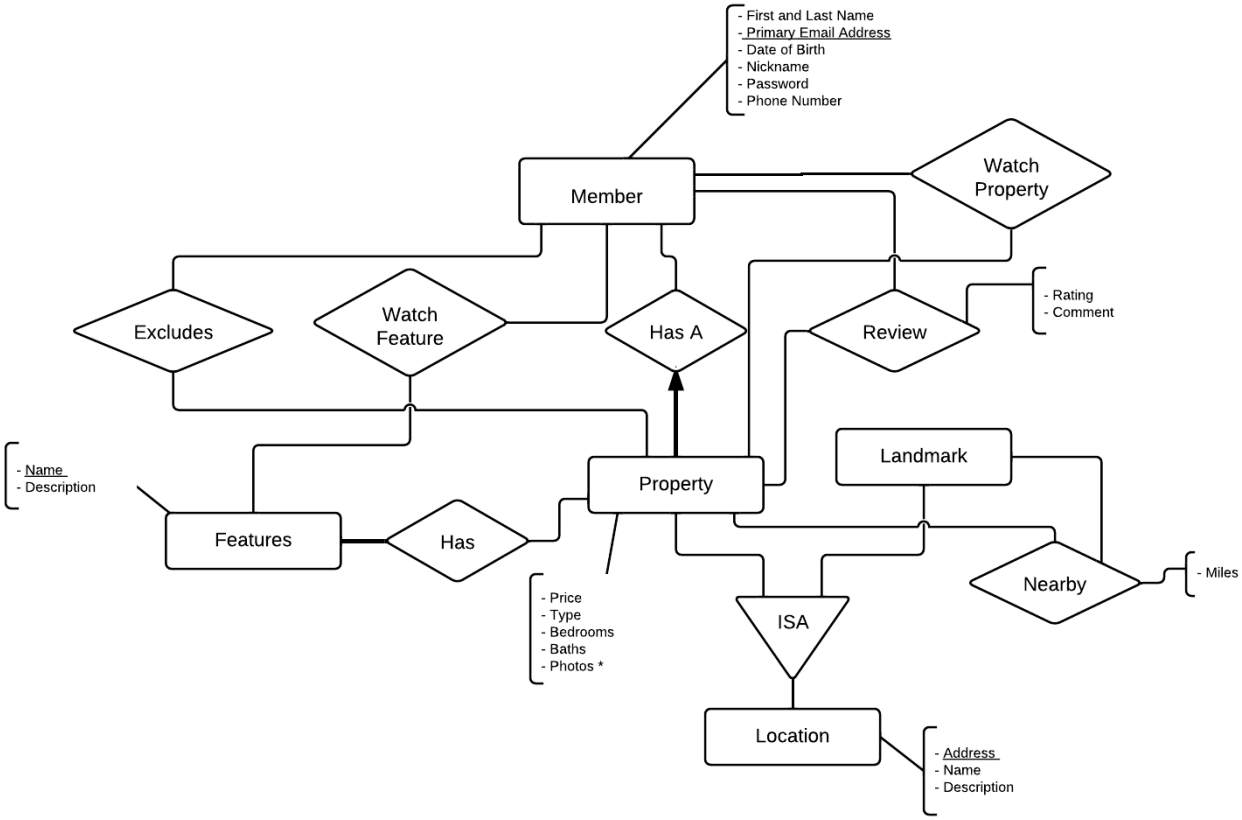
Description: visitors and users should be able to access *BTown Properties* through their mobile phones.

Functional Requirements: *BTown Properties'* mobile version will provide only a limited number of core functions (SF 2.1-6). If the systems detects that the user is using a mobile phone, it should automatically present a mobile version with appropriate presentation styles (CSS) to work on a mobile phone.

Appendix A shows a schematic of the system's features.

3. Data Modeling Design

3.1 ER Modeling Design



3.2 Notes on the ER Model

Only one member owns a property. A member may or may not have one or many properties.

A member may or may not review any number of properties. Likewise, a property may or may not have one or more reviews.

A property and a landmark are kinds of locations. A Property may be near a Landmark.

A member may watch one or many properties or features.

A member may like to exclude properties so that they do not show on search results.

3.3 Relational Schema

Members (**Member ID**, First Name, Last Name, Email Address, Date of Birth, Nickname, Password, Phone Number)

Artificial key Member ID is used for implementation convenience.

Features (**Feature ID**, Name, Description)

Artificial key Feature ID is used for implementation convenience.

The Feature ID represents amenities and is the primary key. For example, a Feature ID with a value of 1 can represent an amenity such as swimming pool, a Feature ID with a value of 2 can represent parking facility etc. The description attribute gives more information about the features.

Has_Feature(**Property ID, Feature ID**)

Property ID is a foreign key to Property ID in Properties entity. Feature ID is a foreign key to Feature ID in Properties entity.

Properties (**Property ID**, Name, Description, Address, Price, Type, Bedrooms, Bathrooms)

Artificial key Property ID is used for implementation convenience. The 'type' attribute indicates whether this property is a rental or sublet. Bedrooms and bathrooms are natural numbers indicating the property's number of bedrooms and bathrooms respectively.

Photos (**Photo ID**, Property ID, Caption)

Artificial key Photo ID is used for implementation convenience. Property ID is a foreign key to Property ID in Properties entity.

Landmark (**Landmark ID**, Name, Description, Address)

Artificial key Landmark ID is used for implementation convenience.

Nearby (**Property ID, Landmark ID**, Miles)

The 'Nearby' relation would give us the distance between a property and a nearby landmark. Property ID is a foreign key to Property ID in Property entity. Landmark ID is a foreign key to Landmark ID in Landmark entity.

Review (**Member ID, Property ID**, Rating, Comment)

Member ID is a foreign key to Member ID in Members entity. Property ID is a foreign key to Property ID in Properties entity. The rating would be scaled from 1 to 5.

WatchListFeatures (**Member ID, Feature ID**)

Member ID is a foreign key to Member ID in Members entity. Feature ID is a foreign key to Feature ID in Features entity

WatchListProperties (**Member ID, Property ID**)

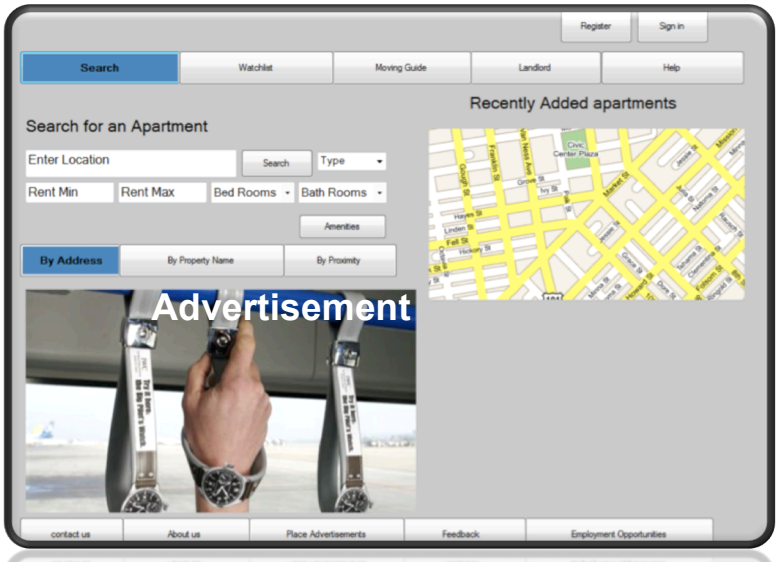
Member ID is a foreign key to Member ID in Members entity. Property ID is a foreign key to Property ID in Properties entity.

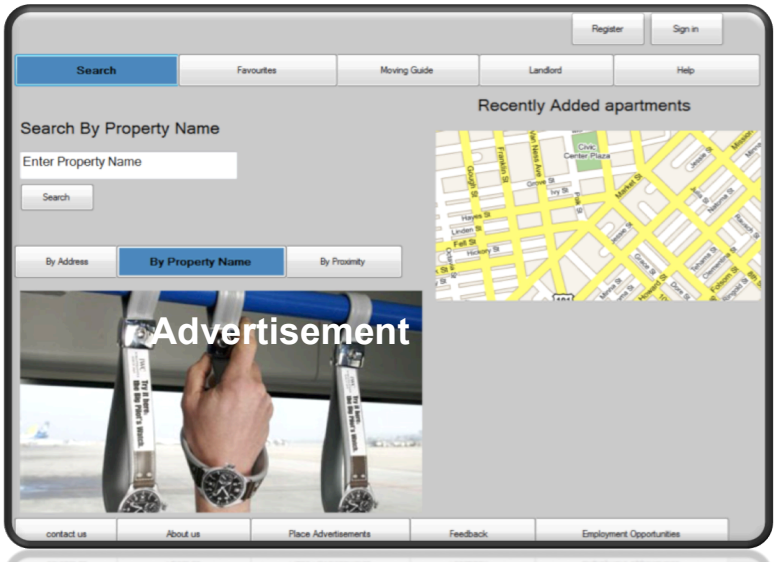
ExcludeProperties (**Member ID, Property ID**)

Member ID is a foreign key to Member ID in Members entity. Property ID is a foreign key to Property ID in Properties entity. Web Interface Design (WD)

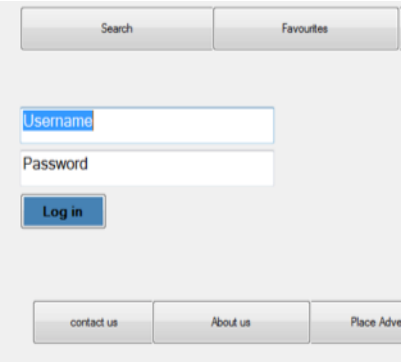
3.4 Web Pages Design

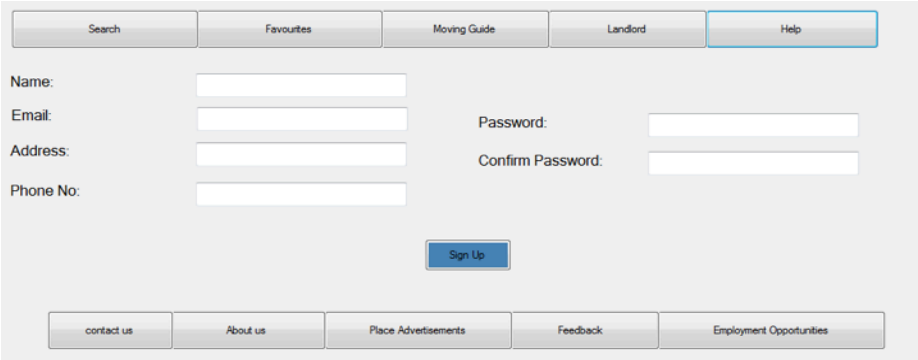
3.4.1 Homepage and Search.



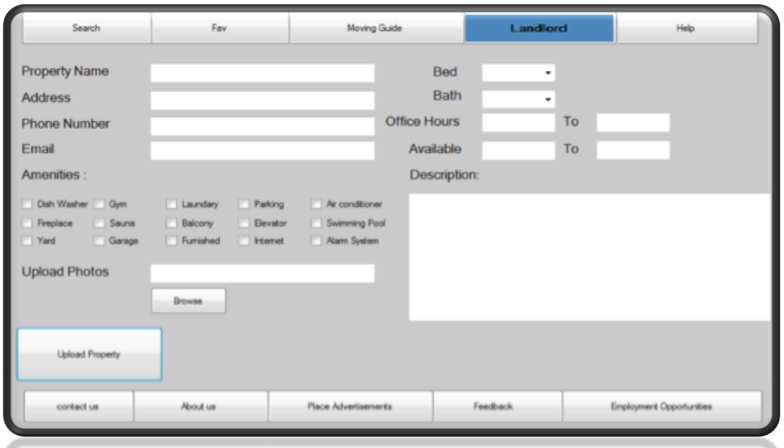


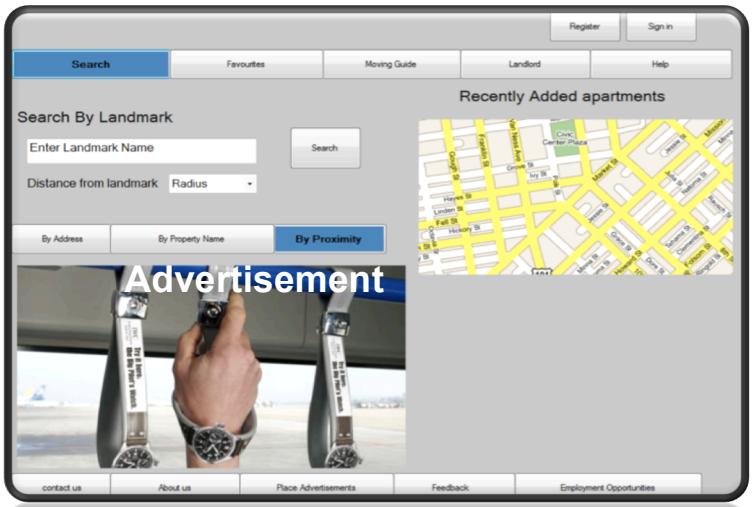
3.4.2 Sign in and Sign up



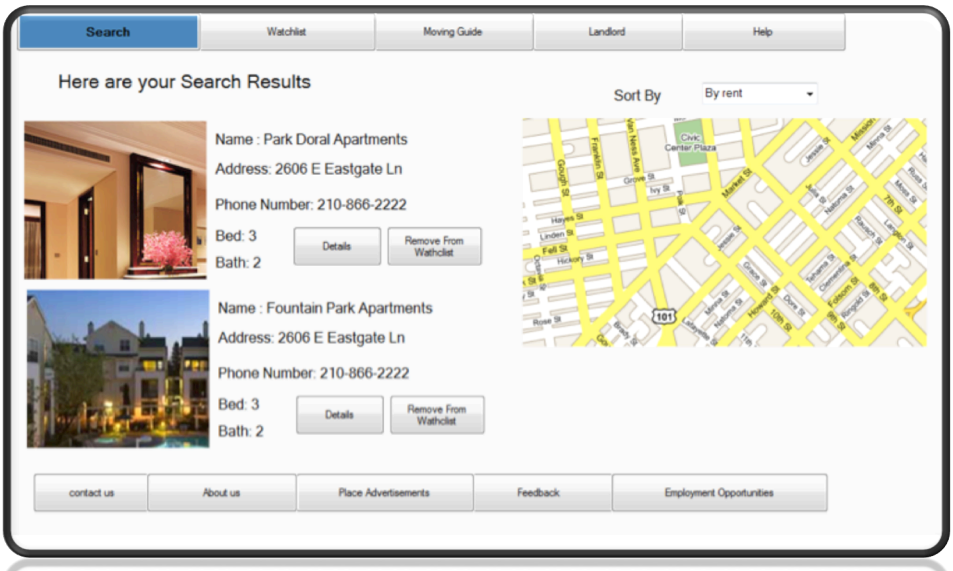


3.4.3 Property Upload Page and Search by Proximity

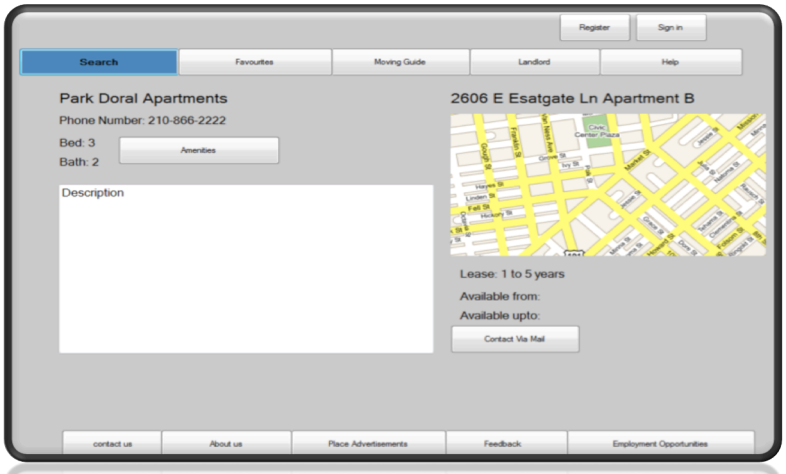


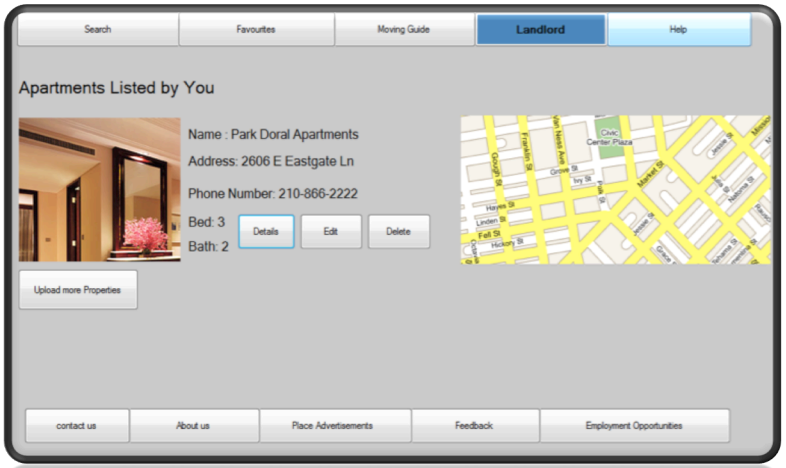


3.4.4 Search results

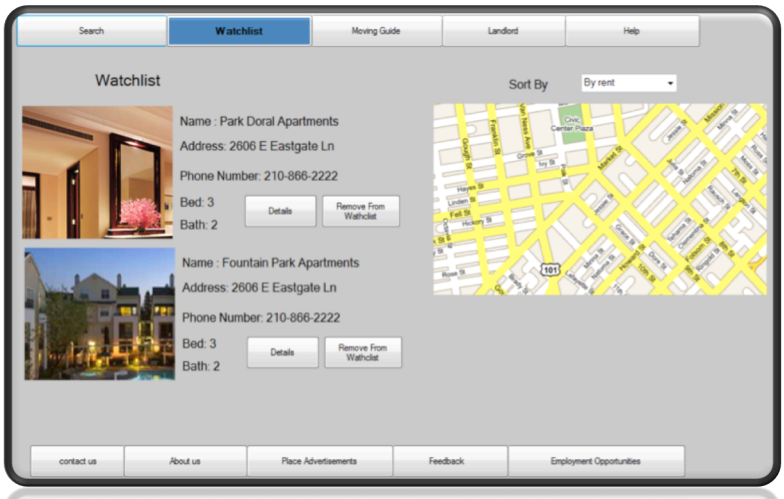


3.4.5 Property Description and listed apartments





3.4.6 Member's Featuers/Properties Watch List Page

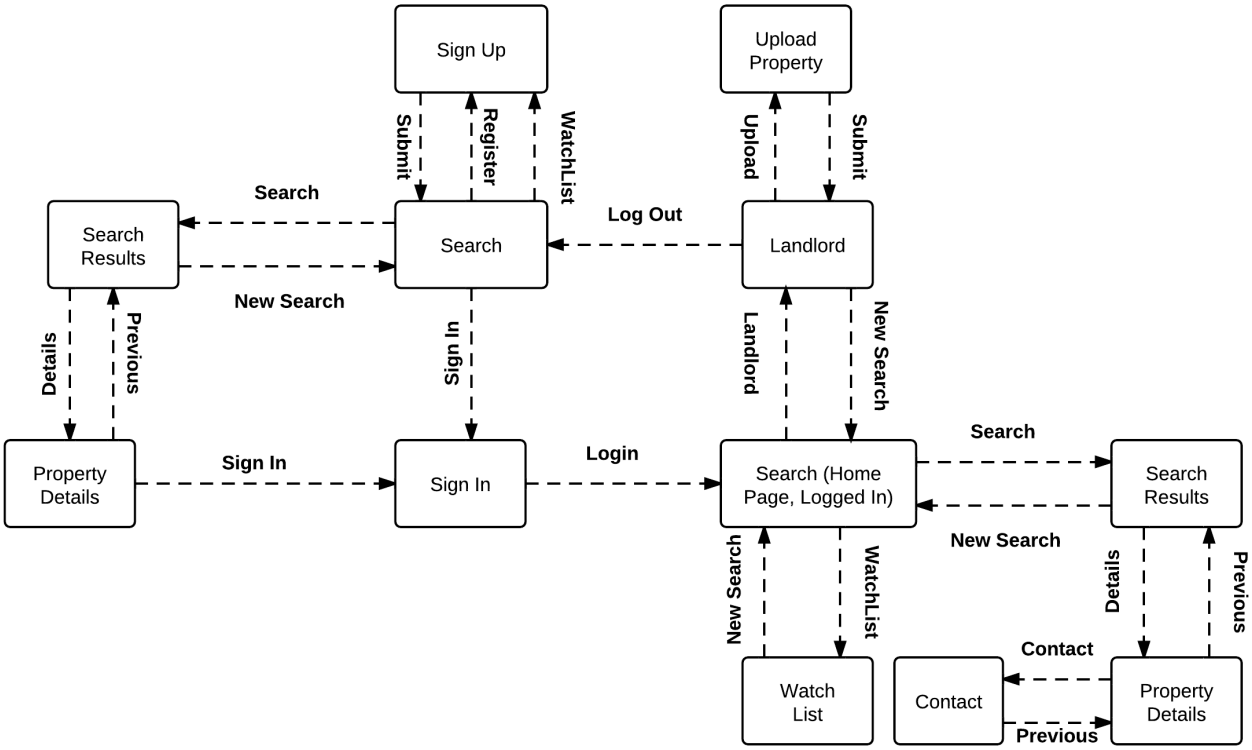


3.5 Transition among the Web Pages

The navigation among the web pages is indicated by the schematic below. Each box represent a Web Page, and each is named for convenience of the reader. A dashed arrow leaving a box represents a click made by a visitor or a member in a page. The arrow points towards the destination reached by a member or visitor.

Note that whenever a visitor clicks on any of the system's restricted features, he is redirected to the sign up page since those features are only accessible to members.

Like discussed in the ER Diagram, visitors can still access all the search facilities, hence a second home page has been created (for visual purposes only) to indicate what a visitor can or cannot do.



3.6 Business Logic

3.6.1 General Rules

A visitor cannot contact a member; only members can interact. When a visitor tries to access restricted resources, the system will redirect him to a sign-up page. The system's restricted resources are: properties upload, contact a member and maintain watch lists.

For privacy reasons, a landlord will not be able to see the members who added his property into their watch lists but will see the count, i.e. the number of users, who added his property into their watch lists.

3.6.2 Specific Rules

The following rules are specific to a web page. We will denote the webpage to which we are referring by its code number; e.g., (WD 4.1.2) refers to sign-in on page 7.

When a user enters a location name (WD 4.1.1), he will be able to see properties listed in that location only. Properties close to the location will not be shown. 'Rent Min' and 'Rent Max' are not mandatory fields, which imply that a user can enter only a 'Rent Min' value or only a 'Rent Max' value to attain preferred results. If visitor or member enters only half of the property name, for example if he enters 'park', the system should be able to display all properties having 'park' in their names such as Park Doral, Fountain Park, etc. The same functionality would apply for any such related searches.

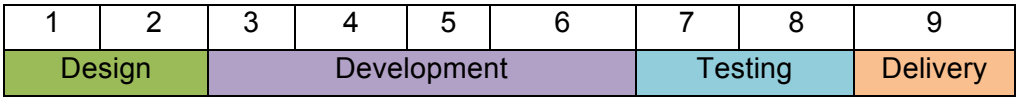
In (WD 4.1.2), if a visitor leaves a field blank or enters an invalid email address, the system has to direct him to correct the information. If the credentials do not match in the system, the visitor will be redirected to (WD 4.1.2), and two messages will be shown: 1) "forgot password?" which will provide functionality to recover a password and 2) "become a member" which will redirect to (WD 4.1.2) Sign Up page. As soon as the visitor registers and becomes a member, a registration email notification would be sent to him to confirm his credentials.

In (WD 4.1.5), in the property description, a "Contact" button will only be shown to members. In the place of the "Contact" button, a visitor will see the message: "Do you want to contact the owner of this property? Click here and become a member to be able to do it"

In (WD 4.1.6) when a member tries to add the same property that he had already added into his watch list, the system will notify him. There can be overlapping between the features and properties watch list.

4. Timeline

The following is the proposed development timeline to develop *BTown Properties.* Each number denotes a week starting on Monday, October 3, 2011; and ending on Friday, December 3, 2011. Then, we detail the activities in a daily basis.



Week	Day(s)	Activity
1	**10/3-8**	Create data model and interface designs based on requirement analysis. Create project proposal document.
2	**10/9-15**	Refine and finalize data model and interface designs, and project proposal document. Create presentation slides for project presentation.
3	**10/16-22**	Create physical database schema. Create interface design template. Install and configure development environment, including framework, group account access, and svn repository.
4	**10/23-29**	Create MVC (Model/View/Controller) scaffolding for each application view, and finalize HTML/CSS for interface design template. Create prototype views and connect global navigation. By end of week, have fully navigable prototype application with blank views.
5	**10/30-11/5**	Incorporate business logic in controllers for the different user roles; public, member, and admin. Create forms for create/modify account, login/logout, rental property entry, and connect to database. By end of week, users can create/modify accounts, login/logout, and create user accounts. Use interface to create test users, sample rental properties, and landmarks.
6	**11/06-12**	Create search, watch list forms/views, and connect to database. By end of week, should be able to run search queries on sample properties, and display results, and add/delete properties from watch list.
7	**11/13-19**	Testing/review cycle. Finish advanced search functionality. Front/back end adjustment/refinement. By end of week, all functionality should be implemented.
8	**11/20-26**	Quality assurance. Create final report and presentation slides.
9	**11/27-12-03**	Finalization, packaging, and delivery of project. Create deliverables; final report, presentation slides, source code.

5. Other Specifications

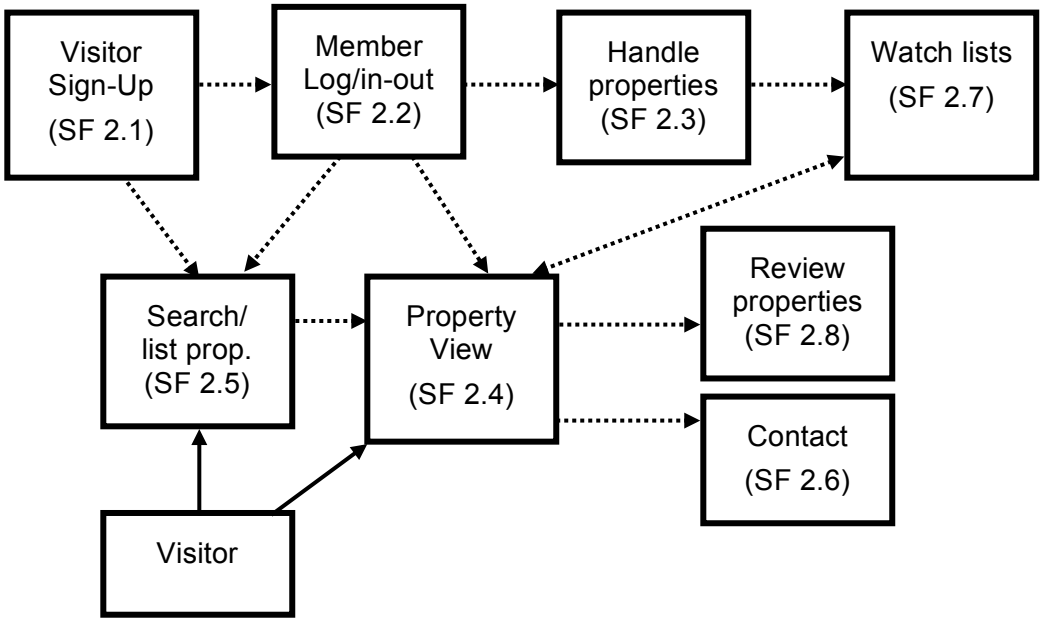
BTown properties will be implemented in PHP 5.3 with MySQL as the database backend. We will develop a custom library of PHP objects to implement the business logic. The business logic and web interface will be integrated into the final application using Zend Framework 1.11, which is built using the MVC design pattern.

Some portions of the application, such as the use of maps to point properties, will use the interface provided by Google Maps.

6. Appendices

6.1 Appendix A: System's Features Schematic and

The interactions among the systems' features are depicted on the *system features schematic design* below. The dashed lines represent operations that can only be performed by members.



6.2 Appendix B: Coding Responsibilities

The following table contains each of our group participants' responsibilities. Each participant is accountable for the correct performance of his part at the end of the semester. Also, each one must make those HTML views associated with their responsibilities and document their code.

Participant	Responsibilities
Gaurav Ranade	SF 2.3 (Upload, Edit and Delete Property) and SF 2.8
Gouri Netravali	Main Layout Design in HTML and CSS, SF 2.4 and 2.6
Jay Mercer	SF 2.5
Ritesh Kavungal	SF 2.7
Suresh Kanagala	SF 2.1 and 2.2. Also, a functionality to edit a member's information and an admin's control panel.
Enrique Areyan	Integrate all parts into the Zend Framework. Session handling. Ensure views' consistency across all pages. Milestone 3 written document.