

I

SW Engineering CSC 648/848 [2] - Fall 2016

“myPlace” Web Application

Team 14 (*Local*)

Sophia Amin - samin25@mail.sfsu.edu

Ilya Ivanenko

Daniel Nguyen

Patrick Aung

Jimmy He

Ulises Martinez

“Milestone 2”

November 1, 2016

Submission Date	History
10/06/16	M1 First Revision
10/11/16	Recommended Revision
10/25/16	M2 First Revision
11/1/16	Recommended Revision 2

Table of Contents

- I. Executive Summary**
- II. Use Cases V2**
 - Guest
 - Student
 - Landlord
 - Admin
- III. Data Definitions V2**
 - Database Schema
- IV. Functional Requirements V2**
- V. Non-Functional Requirements V2**
- VI. UI Mockups and Storyboards (high level only)**
 - Homepage & Login/SignUp (Popup/Tabs)
 - Apartment thumbnail & Full Apartment Description
 - Landlord Page & Add/Edit Apartment (Manage Apartment)
 - Student Page (Manage Favorites)
- VII. Competitive Analysis**
- VIII. High Level Architecture, Database Organization**
 - LAMP Stack
 - Frameworks and APIs
- IX. High Level UML Diagrams**
 - High Level UML class diagrams for implementation of classes
 - UML Deployment Diagrams & Database Schema
- X. High Level APIs**
- XI. Key Risks**
 - Skill Risk
 - Schedule Risk
 - Technical Risk
 - Legal/Content Risk
- XII. Team Organization**

I. Executive Summary

myPlace is an apartment rental web application that is built by San Francisco University (SFSU) students, for SFSU students. *myPlace* leverages itself over its competitors by limiting its rental services to SFSU students only. *myPlace* will include basic features such as search, filters, maps, apartment listing, and communication. However we highlight these features by basing the design similar to official SFSU websites; thereby creating a welcoming and native feel familiar to SFSU students as well as an extension of SFSU's online identity. By emphasizing simplicity and intuitive design, we can serve clients of all ages and focus on delivering the one desire of our SFSU students: finding a place to rent.

Guests and students shall be able to view apartment listings provided by landlords and filter results to better find exactly what they are looking for. After finding an appropriate living space, the student can contact the landlord to get more information and setup appointments.

Landlords have the ability to create apartment listings that can be seen by both guests and students. Landlords can provide information such as images, address, available times, and etc, which will allow students to better filter for their own apartment.

All postings shall be moderated by the sites administrators who can remove inappropriate or no longer relevant apartment listings. Administrators can also remove user accounts that are unproductive or toxic for the website's community.

myPlace's purpose is to deliver a simple and reliable apartment renting service to SFSU students. Although this type of service is not new, there has yet to be a renting service that targets SFSU students adequately.

As a small startup of SFSU students, we believe that *myPlace* will not just provide a better renting service, but also foster a better relationship between SFSU students, SFSU itself, and local landlords. We know the difficulties and obstacles in finding the perfect apartment and can provide competitive resources so that each user has the best overall experience.

II. Use Cases V2

2.1 Gary the guest - Gary is looking for **apartments** in San Francisco and stumbles upon this website. Gary types in “one bedroom pet friendly” in the search bar and is impressed at how easy it is to filter through the listings to find his ideal apartment. He clicks on the apartment thumbnail only to find out that he must register on the website as a SFSU student in order to contact the landlord. Gary remembers that he has a niece named Sam who will be transferring to SFSU and recommends the website to her.

2.2 Sam the student - Sam is a new transfer student at San Francisco State University. She will be attending the university to complete a bachelor's degree in Holistic Health. Since she will be attending school for two years, she wants to find an apartment where she can spend the semester then move back home with her parents during the holidays. As a student, she would like to search for apartments before every semester and find locations that are not too far away from campus. She enjoyed this site's ease of use and features including filters and secure ways of communicating with landlords.

2.3 Leah the landlord - Leah just found out on short notice that she is leaving on a seasonal business trip in a few days. She wants to rent her apartment next to SFSU temporarily to a student. Due to her limited time, she needs to post an apartment listing and find someone immediately. She particularly liked how easy the website made it to describe what her apartment offers. This saved her time by giving her the option to know exactly what type of information she needs to provide.

2.4 Andy the Admin - Andy is logged in as the website admin. Andy is browsing through new apartment listings and notices that some images uploaded by the landlord contain inappropriate content. Andy then deletes the apartment and contacts the landlord to give him/her penalty. While this was happening, Andy receives a notification due to a flagged content by a user. Andy reviews the flagged content to determine whether or not it breaches the website's policy. If so, Andy will remove the apartment and contact the landlord to give him/her a penalty. If not, then Andy will simply remove the flag. Afterward, Andy will repeat these operations.

III. Data Definitions V2

- 3.1 **Guest**: An unregistered user who can browse through apartments without having to register. Has limited view of apartment information.
- 3.2 **Student**: A registered user that can browse through apartments and contact landlords for information. Has more access to information than guests.
 - 3.2.1 **Favorite** apartments to save for later
- 3.3 **Landlord**: A registered user that can post apartment listings for rent by uploading basic information.
- 3.4 **Admin**: Can moderate and delete landlord's' apartment or accounts
- 3.5 **Apartment**: A location that can be posted by landlords up for rent and viewed by guests or students.
 - 3.5.1 **Image**: Image used to display on website. (apartment Pictures, thumbnail, etc) stored as BLOBs
 - 3.5.2 **Rating**: Number of people that added the apartment to favorites
 - 3.5.3 **Date Posted**: The date the apartment listing was created
 - 3.5.4 **Title**: Attractive explanation by landlords about apartment
 - 3.5.5 **Description**: Description about apartment provided by landlord
 - 3.5.6 **Map+Bus Routes**: Map containing general location of apartment along with bus routes using Google API (not stored in database)
 - 3.5.7 **Thumbnail**: Image and concise description of apartment
- 3.6 **Filters**: Refines user's search to find the most appealing apartment
 - 3.6.1 **Directions Service**: Provided by Google API
 - 3.6.2 **Distance from SFSU**: Calculated by Google API
 - 3.6.3 **Parking availability**: Displays if parking is available
 - 3.6.4 **Price Range**: Input a price to narrow down search
 - 3.6.5 **Rental Term**: The availability of the apartment (eg. January-May)
 - 3.6.6 **Bedrooms**: Number of bedrooms available
 - 3.6.7 **Pets**: Displays apartments that allow pets inside
 - 3.6.8 **Zip Code**: Zip code of apartment location
- 3.7 **Tags**: A descriptive feature about an apartment used to narrow searches and enhance filter usage
 - 3.7.1 **Predefined tags**: Provided by website and are selected upon listing creation
 - 3.7.2 **User Defined tags**: Made by landlord as listings are created
- 3.8 **Search Bar**: A search field where the user may type in keywords that they are searching for in an apartment
- 3.8 **Sort By**: Arranges apartment results by different sorting methods
 - 3.8.1 **Price**: High to Low
 - 3.8.2 **Price**: Low to High

IV. Functional Requirements V2

Priority Level: (1 - *must have*; 2 - *desired*; 3 - *opportunistic*)

<u>Priority 1:</u>	
Data Definition	Function
<u>Website</u> shall have:	Use of filters, search bar, and sort by features
	Login and SignUp
	Thumbnail of Apartments & Featured Apartments
<u>Guest</u> shall be able to:	Register for an account (only as SFSU student or Landlord)
	Browse apartments
	Filter, search, or sort apartment results
<u>Student</u> shall be able to:	Login account with mail.sfsu.edu email
	Filter, search, or sort by apartment results
	Communicate with landlords
	Message Landlord
<u>Landlord</u> shall be able to:	Offer (multiple) apartments for rent
	Edit apartment (ie. description, images, delete, etc.)
	View messages from interested students
	Have guest view of apartments
<u>Admin</u> shall be able to:	Use SQL Workbench to delete user accounts and/or apartments but not edit them
<u>Apartment</u> shall have <u>data</u> including	Title
	Description
	Price
	Zip Code
	Maps using Google Maps API

	Images (optional - max 10)
	Thumbnail
	Posted Date
	Pet Friendly
	Parking Available
	Laundry Available
	No Smoking
	Shared Room
	Furnished
	Wheelchair accessible

<u>Priority 2:</u>	
Data Definition	Function
<u>Website</u> should be able to:	Sort apartments by options such as rating, posting date, etc.
	Iconography for common UI elements
<u>Student</u> should be able to:	Save favorite apartments
<u>Apartment</u> should have <u>data</u> including:	Tags
	Distance from SFSU calculated using Google APIs

<u>Priority 3:</u>	
Data Definition	Function
<u>Student</u> <i>may be able to:</i>	Communicate with landlords using built in chat

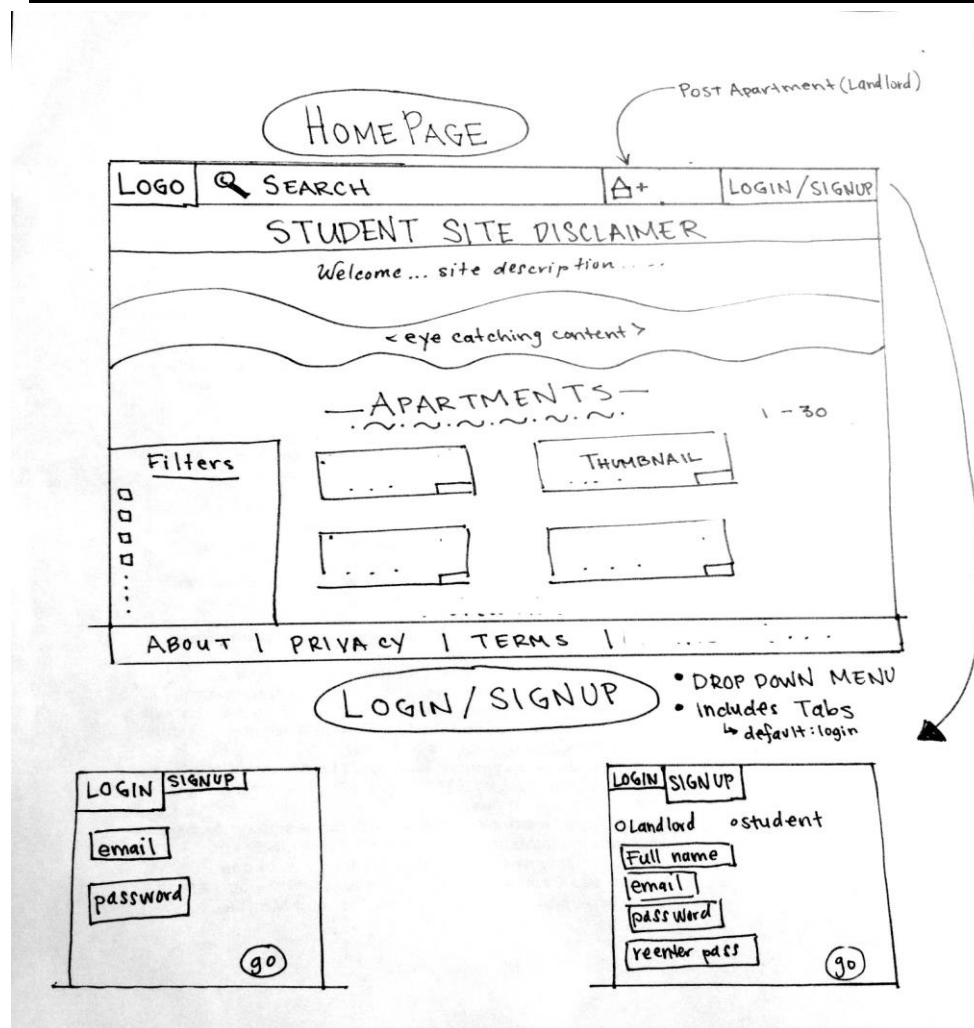
<u>Landlord</u> may be able to:	Option to bump listing to the top of the list of apartments (after 7 days)
<u>Apartment</u> may have <u>data</u> including:	Bus Routes to SFSU using Google APIs
	Rating

V. Non-functional Requirements V2

1. Application shall be developed using class provided LAMP stack
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks shall be explicitly approved by Marc Sosnick on a case by case basis.
 - We shall use the following approved tools: JQuery Bootstrap, PHP, MySQL Workbench, Netbeans.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class.
4. Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. It shall degrade nicely for different sized windows using class approved programming technology and frameworks so it can be adequately rendered on mobile devices.
 - The application shall utilize JQuery to target Bootstrap's predefined CSS classes to customize our web design.
 - The application shall utilize responsive web design features provided by the Bootstrap framework to maintain the application's design layout.
5. Data shall be stored in the MySQL database on the class server in the team's account
6. Application shall be served from the team's account.
 - The application shall be deployed on the sfsuswe.com server, under the account of f16g14.
7. No more than 50 concurrent users shall be accessing the application at any time
8. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
 - Landlord apartment listing details shall be hidden to guest users.
 - The privacy policy shall be presented to the each user upon registration and it must be accepted before an account is created.
9. The language used shall be English.
10. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

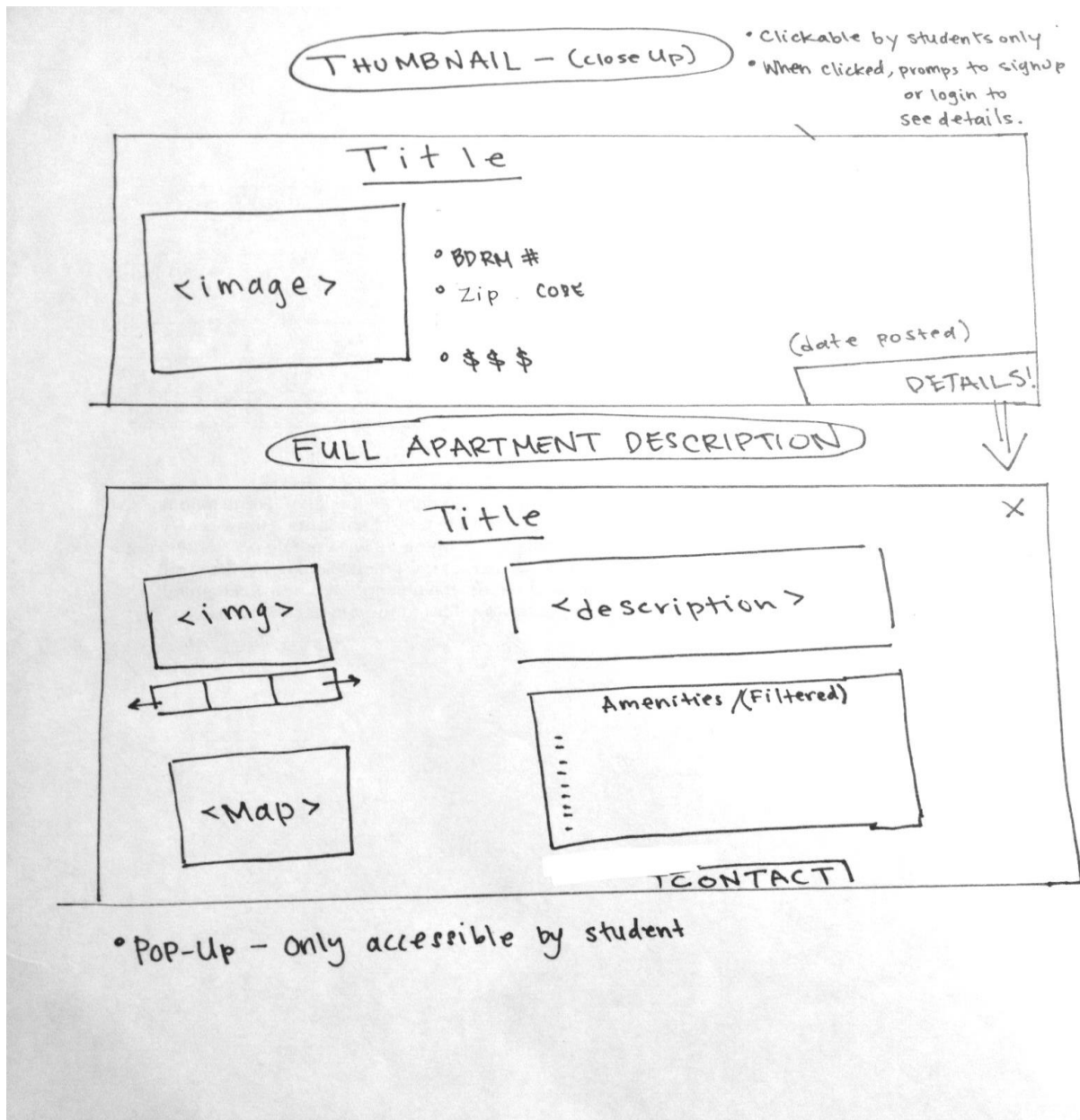
- The application shall follow design properties of official SFSU websites, providing users who are familiar with SFSU to quickly grasp our design.
- 11.** Google analytics shall be added for major site functions.
- 12.** Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.
- 13.** Pay functionality (how to pay for goods and services) shall be simulated with proper UI, no backend.
 - The Payment interface shall have common transaction items such as usernames, pay amount, payment type (e.g credit card details), and the goods exchanged. A proper receipt shall be displayed after completing the transaction.
- 14.** Site security: basic best practices shall be applied (as covered in the class)
- 15.** Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development, and only the tools and practices approved by instructors.
 - Group 14 shall use email, Google Drive, GroupMe (texting service) for collaboration and feedback.
- 16.** The website shall prominently display the following text on all pages
“SFSU/FAU/Fulda Software Engineering Project, Fall 2016. “For Demonstration Only”. (Important so as to not confuse this with a real application).

VI. UI Mockups and Storyboards (high level only)



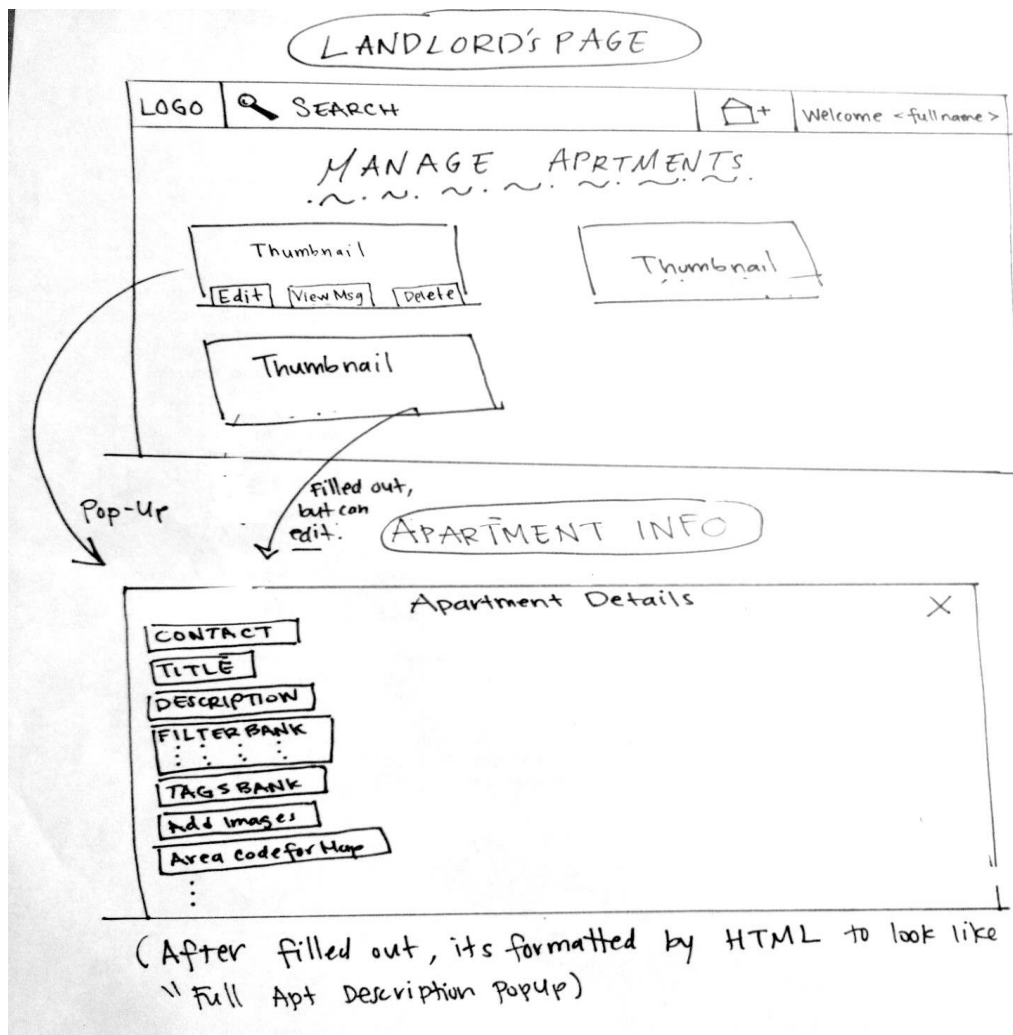
Homepage for myplace

Guests, students, and landlords will come to the homepage to view all the apartment listings. This page is layed out where anyone can look at the apartments available and **filter** and **sort** the results to refined it to their liking. In addition, they can also use the **search engine** to type in keywords or tags they are looking for in an apartment. Users will only see the **thumbnail** of each apartment. In order to see the full description, users must be a registered student user. From the homepage there are several places where users can navigate to and will be prompt to sign up or login for full access. There are three tabs which will do this by the standard login/signup button, post apartment button, and view favorite apartments button (priority 2). The **login/signup** button will be a pop up, while the **post apartments** button will open to a new page for landlords to manage.



Thumbnail (close-up from homepage) & Full Apartment (popup)

The thumbnail will be viewed by all users on the homepage, where they will see a brief representation of the apartment. Only registered student users can click on the “see details” button to view the full apartment description pop up. Clicking on the **star** allows students to add that specific apartment to their favorites. If a user who is not registered clicks on it, they will be prompt to register as a student to use the feature. (Priority 2)



Landlord's page & Apartment Info (popup)

The **landlord's page** is where landlords can manage their apartment listings called **Manage myApartments**. They will be allowed to post multiple apartments under one account which is handled in the same location. The **Add New** button pops up a new apartment application where the landlords can fill out all the information they would like to provide to students about the apartment. The mandatory information needed is what is shown in the *thumbnail* (i.e. *rent term, zip code, price, number of bedrooms, and distance from school*). If landlords want to edit their previously added apartment, they can hit the **edit** button on the thumbnail which will also open to the apartment application with their previously saved information to edit.

VII. Competitive Analysis

Features	places4students	sfstate.renthello	Our Future Product
Map	+	+	+++
Search	+	+	-
Member Login	+	+	+
Ease of Use	+	+	+++
Security	+	+	+
Filters	+	+	+++

+ Feature exists; +++ superior; - does not exist

myPlace's closest competitor is sfstate.renthello.com (RentHello). RentHello is another apartment renting web application that caters to SFSU students. However, anyone may register to RentHello without verification. *myPlace* will verify each renter as an SFSU student. Also, RentHello follows the design of the official SFSU website, but are convoluted due to its dynamic content and excessive styling. *myPlace* will display only what is needed to give SFSU students a familiar feel to the official SFSU main website and serve their needs.

VIII. High Level Architecture, Database Organization

The following tools, APIs, and frameworks will be used to assist the development of this web application.

8.1 LAMP Stack

The LAMP stack is a very popular web platform that allows for easy setup and maintenance of web applications. It's acronym stands for Linux, Apache, MySQL, and PHP which are used together to provide all the tools needed for developing and hosting web applications.

8.1.1 Linux is an open source operating system on which the rest of the LAMP stack runs.

8.1.2 Apache is the component of the LAMP stack that acts as a web server. It processes all incoming/outgoing HTTP requests/responses and, like Linux, is open source as well.

8.1.3 MySQL handles communication with the database for the server. It allows for multiple users to access, create, and modify database entries and uses SQL for these communications.

8.1.4 PHP is the programming language we will use to develop our webpages. PHP can be embedded easily into HTML which makes it easy to use for web development.

8.2 APIs and Frameworks

APIs and frameworks go hand in hand when developing web applications. They provide a set of tools and libraries to utilize for the creation of web applications.

8.2.1 Mini is a simple open source framework made specifically for PHP. It follows the traditional MVC (Model - View - Controller) architectural pattern.

8.2.2 Google Analytics API will allow us to collect, configure, and report on user interaction.

8.2.3 Google Maps API contains methods to display the apartments on a map so users can visually see where apartments are located. In addition, "Directions Service" will be enabled with the mode set to TRANSIT to offer possible public transportation routes to SFSU.

8.2.4 jQuery is an open source JavaScript library designed for the client-side scripting of HTML. This will allow for the creation of dynamic web pages.

8.2.5 Bootstrap is a front-end framework for designing websites and web applications. It combines HTML, CSS, and JavaScript into this one framework for ease of use.

8.2.6 Supported Browsers that our website will support are the latest two versions of the following browsers:

8.2.6.1 Safari: 10.0 and 9.1.3

8.2.6.2 Chrome: 54.0.2840 and 53.0.2785

8.2.6.3 Mozilla: 49.0.1 and 49.0

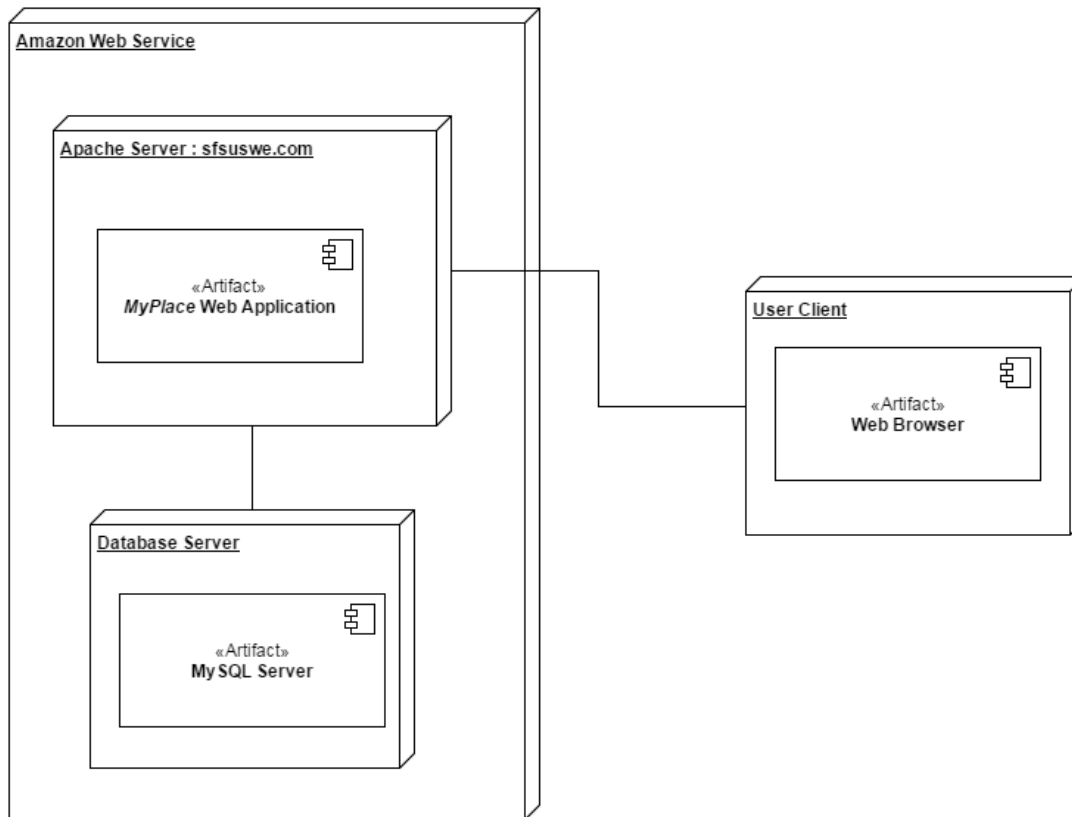
9.1 Database Search Algorithm: The database will be searched using "WHERE column LIKE '%term%'" from the SQL query. After retrieving a search term from the user the function will parse each individual term and pass it to a SQL query in the database. Using "WHERE column LIKE '%term%'" the database will retrieve the apartments that have those keywords. The query will be applied to specific columns of the Apartment table such as the tags column which will be used to search for keywords.

IX. High Level UML Diagrams

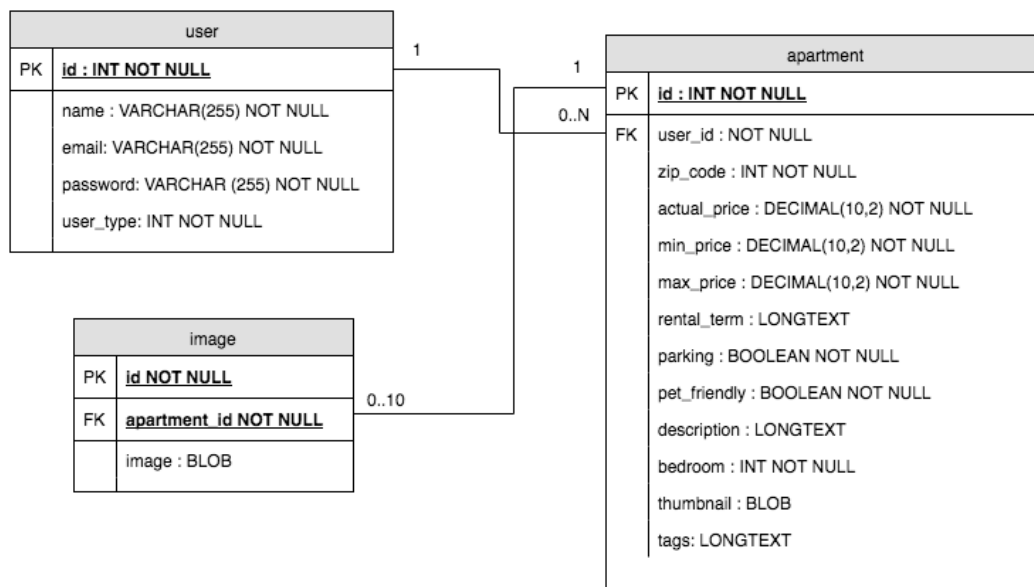
9.1 Class Diagram



9.2 Deployment Diagram



9.3 Database Schema



X. High Level APIs

- Controller

10.1 Page Template: All the functions implemented in the template page will extend to the Home, Student, and Landlord page. This is responsible for retrieving information from the apartment and user databases and displaying on pages.

10.1.1 formatApartment(): Takes data retrieved and formats apartments on page

Return: Formatted html string

10.1.2 search(query, filters): Takes user input to search keywords in web application and formats them to pass to database function call.

10.1.3 logout(): Logs out current user

10.1.4 login(user): Logs in the user (Landlord or SFSU student)

10.1.5 checkLogin(): Checks if a user is logged in

Return: Int (based on user type)

10.1.5 displayUserName(): Displays the name of logged in user

Return: String

10.2 Home Page: This is the landing page all users will see regardless of if they're logged in or not.

10.2.1 signUp(): Add user (student or landlord) to the user database

10.2.2 applyFilters(): Apply user's selected filter preferences and search results of apartments

10.3 Landlord Page: This is the page landlords will be able to access which includes management of their apartments.

10.3.1 addApartment(apartment): Adds apartment to the database

10.3.2 editApartment(apartment): Edits apartment in database

10.3.3 deleteApartment(apartment): Deletes apartment from database

10.3.4 displayLandlordApartment(): Displays landlord's apartment to page

Return: formatted html string

- **Library**

10.4 Apartment: An apartment is what will be available to rent. Among other things, each apartment will have basic information which will include the zip code of where it is located, the price, how long is the rental term for and how many bedrooms there are.

10.4.1 createMap(int): This function will create a map for the user from the school location to the general location of the apartment.

10.4.2 apartment_id(int): Database generated ID for apartment

10.4.3 zipCode(int): Holds the zip code of apartment

10.4.4 priceRange(double): Range of price for apartment

10.4.5 Image(String): Image stored as BLOBS

10.4.6 parking(boolean): Yes or No

10.4.7 petFriendly(boolean): Yes or No

10.4.8 description(String): Optional description of apartment

10.4.9 thumbnail(String): Stored as BLOBS and formatted HTML

10.4.10 bedroom(int): Count of bedrooms

10.4.11 tags(String[]): Predefined or User defined tags

10.5 User: This class represents a general user object that contains information about a user to send amongst classes. This includes username, email, and password.

- **Model**

10.6 UserDB: This database will contain information about all registered users

10.6.1 addUser(user): add an user to the database

10.6.2 findUser(user): will check if user already exists within database
Return: Array

10.7 ApartmentDB: This database will contain information about every apartment listed

10.7.1 addApartment(apt): add an apartment to the database

10.7.2 deleteApartment(aid:int): delete an apartment from the database

10.7.3 editApartment(apartment): Edit information of apartment

10.7.4 getLandlordsApartment(userID): Retrieves all apartments belonging to a landlord from the database.
Return: String array

10.7.5 search(query, filters): Query the database for apartments that fit the query terms and filter settings.
Return: String array

XI. Key Risks

- 11.1 Skill risk:** The first few milestones allowed us to be familiar with web development but further into the development may pose as a challenge for us when we put the moving parts together. To resolve this deficiency the team will study a lot of online material in forms of YouTube videos and tutorials to support the information we need. We will also use Google and Stackoverflow to help of resolve particular issues that we run into.
- 11.2 Schedule risk:** Due to various work and school scheduling it is hard to have all six members present at the same time. We are able to schedule at least one day before class but this might pose as a problem for later on. We will work to have at least two days of teamwork and about 2 hours for each session. So far this has worked for the first two milestones.
- 11.3 Technical risk:** We are still debating on whether or not to use BLOBS for our images in our database and weighing its technical advantages and disadvantages to suit better our needs. We are all researching how to implement this.
- 11.5 Legal/content risk:** We will be careful and be getting our images from a website that does not have any copyrighted pictures

XII. Team Organization

Member	Role
Sophia Amin - samin25@mail.sfsu.edu	Team Lead, Front-End
Ilya Ivanenko	Team Tech Lead, Controller
Daniel Nguyen	Back-End Developer
Patrick Aung	Database Manager
Jimmy He	Back-End Developer
Ulises Martinez	Front-End Developer

12.1 Responsibilities:

- 12.1.1 Team Lead** establishes teamwork, organize & schedule meetings, and help setup collaborative & development tools.
- 12.1.2 Tech Lead** is in charge of key technical decisions. As well as helps and educates team with programming and architecture.
- 12.1.3** The **Controller** will send commands to the model to update the state of the MVC model.
- 12.1.4 Front-End Developer** will be designing the user interface and creating the responsive layouts for **user experience** with scripts embedded in the Web application.
- 12.1.5** The key role for the **Back-End Development** is to process and pull requests from the database, format the data, and send it back to the view.
- 12.1.6 Database manager** will be developing and maintaining the architecture of the database.