SW Engineering CSC 648/848 Spring 2019



Team 02

Jaren Lynch
Chris Rosana
Monali Mirel Chuatico
Thanh Le
Shan Kwan Cho
Russelle Pineda
KaHo Lee

Milestone 2

History Table

Progress	Date
First Draft	March 21, 2019

Table of Contents

- **I. Data Definitions V2**
- II. Functional Requirements V2
- III. UI Mockups and Storyboards
- IV. High level architecture, Database Organization
- V. High-level UML Diagrams
 - A, Class Diagrams
 - **B. Component and Deployment Diagram**
- VI. Key Risks
- VII. Project Management

1. Data Definitions V2

- 1. Unregistered users: The user who comes to the site and looks for a room to rent.
- 2. Registered users: the user is registered and shall receive more benefits than unregistered users. To be a member of "Stay", this person must provide a valid contact information such as phone, email, and/or optional profile photo. These requirements not only to ensure that the user isn't fraudulent but also help them easily connect to the website.
- 3. Administrator: In general role, this person mainly interacts with users. This person has the most authority to track all users' activities. In order words, this person is a supervisor of all customer transactions.
- 4. Developers: A person who analyzes and develops the website. This person shall continue working on the site to maintain and debug the page when needed
- 5. Listing: different and connected items that users can view to manage their searches. Subitems listing has location, housing type, price, etc.
- 6. Login: identifier user's information on Stay's system. It requires User ID and password.
- 7. Sign-up: authorize a user to be a member on this site.
- 8. Search: the results returned from querying for listings.
- 9. Saved listing: listings that users want to check back later on
- 10. Tenant: a registered user who looks for a place to rent.
- 11. Landlord: a registered user who looks for a person needing a place to rent out.
- 12. Transaction: The booking of a rental property.
- 13. Registration record: recording all user registration data such as names, emails, phone numbers, address, etc.
- 14. Posting-item: where users post housing information for rent. This item contains housing type, description, price, location, contact information, etc.
- 15. Image: photo of housing type.
- 16. Price: the price tag to be paid for rent.

- 17. Description: describe the condition of the house.
- 18. Categories: list of items categorized in three different groups: apartment, house, or room. This makes the searching tool faster and more effective.
- 19. Search: tool where users use to find information quickly.

2. Functional Requirements V2

Priority 1:

- 1. Unregistered user
 - 1.1. Unregistered users shall be able to view the website.
 - 1.2. Unregistered users shall be able to login or register.
- 1.3. Unregistered users shall be able to use searching tool to find housing types they are looking for.
 - 2. Registered user
 - 2.1. Registered user shall be able to do anything unregistered user can do.
 - 2.2. Registered users shall be able to upload a listing on the site.
 - 2.3. Registered users shall be able to log out of their accounts.
 - 3. Administrators:
 - 3.1. Shall be able to contact all customers using this site.
 - 3.2. Shall be able to ban registered users.
- 3.3. Shall be able to check and approve or remove registered users' posts before they upload on the site..
- 3.4. Shall be able to view historical data such as searches, making payment, listings, etc. from all registered users.
 - 4. Developers
 - 4.1. Developers shall be able to edit the site.
 - 4.2. Developers shall be able to view databases and maintain the server.

Priority 2:

- 1. Unregistered users:
- 1.1. Unregistered users shall provide username, email, and password to register to this site.
 - 1.2. Unregistered users shall be able to sort listings.

- 1.3. Unregistered users shall be able to cancel creating a new account while in progress.
 - 2. Registered users:
 - 2.1. Registered users shall be able to edit their posts.
 - 2.2. Registered users shall be able to view their own historical activities.
 - 2.3. Registered users shall be able to post photos to their profiles.
 - 2.4. Registered users shall be able to contact landlords and tenants.

Administrators:

Shall be able to track each transaction.

4. Developers:

- 4.1 Developers shall be able to edit the site.
- 4.2 Developers shall be able to view internal and external interfaces.

Priority 3:

- 1. Unregistered users:
 - 1.1 Unregistered users shall be able to login or register.
 - 1.2 Unregistered users shall be able to sort listings.
 - 1.3 Unregistered users shall be able to search by location, price, type, and distance to campus.

2. Registered users:

- 2.1 Registered users shall be able to change their information.
- 2.2 Registered users shall be able to upload a listing on the site.
- 2.3 Registered users shall be able to favorite a listing.
- 2.4 Registered users shall be able to view their own historical activities.

3. Administrators:

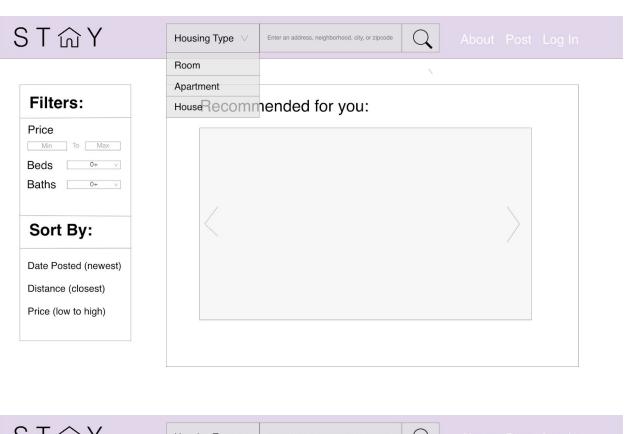
- 3.1 Shall be able to track each transaction.
- 3.2. Shall be able to contact all users

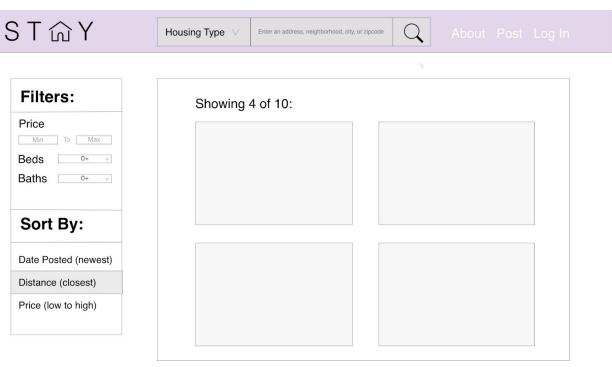
4. Developers:

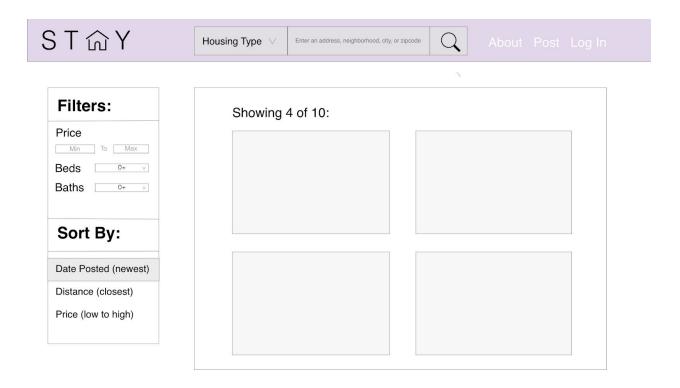
- 4.1 Developers shall be able to edit the site.
- 4.2 Developers shall be able to view databases and maintain the server.
- 4.3 Developers shall be able to view internal and external interfaces.

3. UI Mockups and Storyboards

1. User searches for housing using filters and sorts:



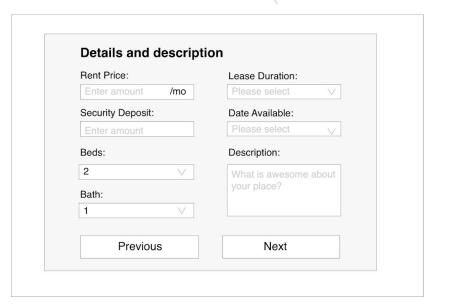


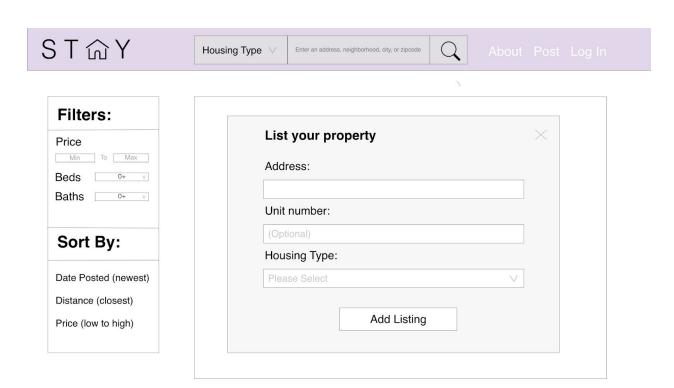


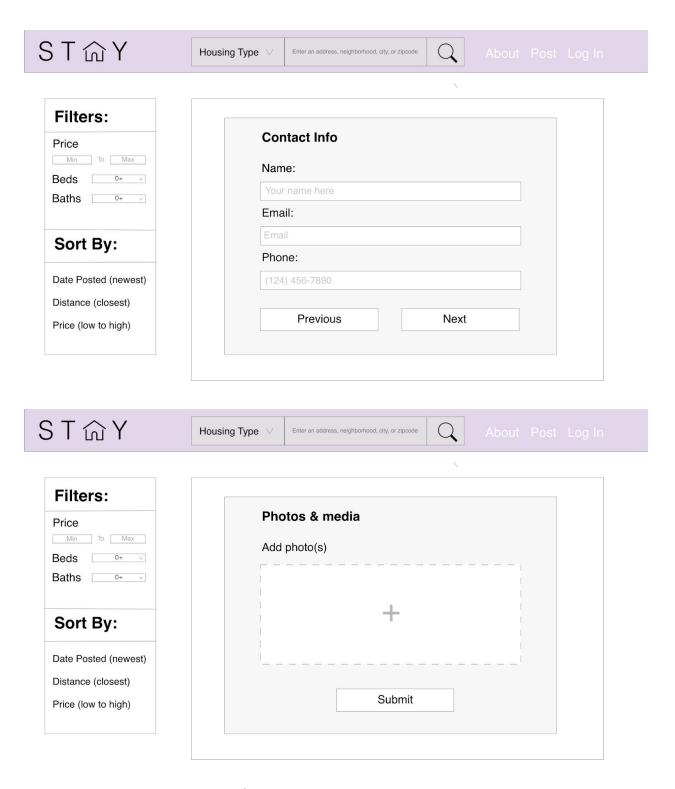
2. User posts a new listing:



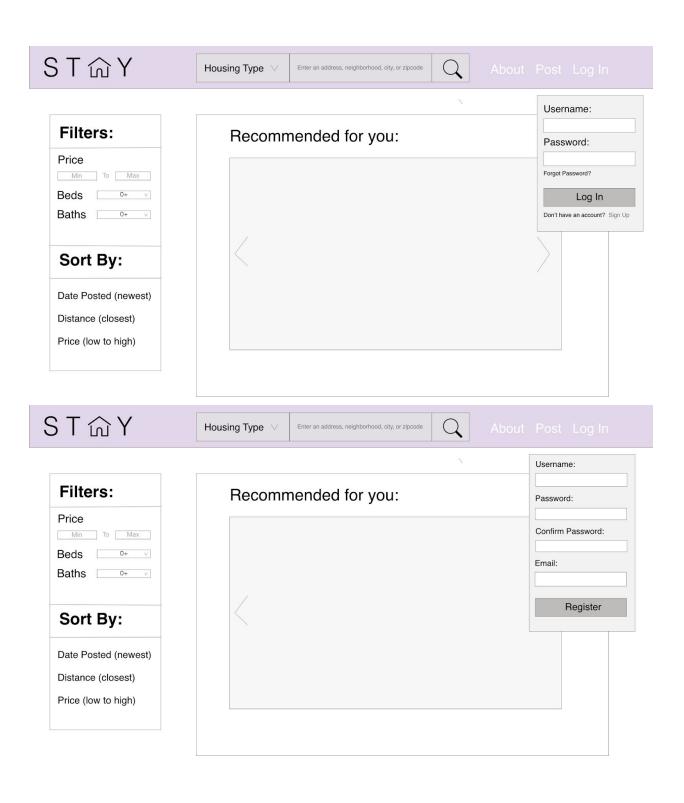








3. User is able to log in or sign up for an account:



4. High level Architecture, Database Organization

- DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables (e.g. their titles) and items in each DB table (check instructors' suggestions and class slides on architecture). Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above.
- Media storage: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made by the end of M2). Describe any other special data format requirements like for video/audio/GPS etc.
- · Search/filter architecture and implementation: what will be the alg/SW for search; how will you organize search items for the user; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class. OK to use SQL and %like).
- · Your own APIs (if any): Describe and define at high level any major APIs that you will create other than standard ones provided by tools and frameworks you use
- Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)
- If you have changed SW tools and frameworks or added any new one please describe it. Any new SW or framework you will be using has to be approved by CTO in writing by this time.

Database Organization

-User

- Username ID (PK)
 - The unique username account.
- Name
 - Name information of the user.
- Email
 - Email address of the user.
- Password
 - o Password for the user account. Personally created by the user.
- Phone number
 - The phone number of the user.

-Landlord

- Landlord ID (PK)
 - The unique landlord account.
- Name
 - Name information of the landlord.
- Email
 - Email address of the landlord.
- Password
 - o Password for the landlord account. Personally created by the landlord.
- Phone number
 - The phone number of the landlord.

-Posts/Listings (Home/Room/Apartment)

- Listing ID (PK)
 - Unique listing ID.
- Title or ID
 - Most likely the address of the listing.
- Type
 - The type of listing whether it is a house, room, or apartment.
- Price
 - A monthly price of the listing.
- Description
 - A detailed description of the listing provided by the landlord.
- Date posted
 - The date the listing was posted by the landlord.
- Availability
 - The status of the listing if it is still up for rent, or it has been rented already.
- Images
 - Images of the listing posted by the landlord.

-Categories

- Category name
 - The name of this category.
- Unique ID
 - This unique ID will be what the foreign key value in a post refers to, tying a post to a given category.

-Message

- User
 - Foreign Key referring to the user who owns a given message so it can be loaded into user account page.
- Sender
 - o The user who sent a given message.
- Receiver
 - the user who receives the message.
- Date
 - o The date when the message was sent or received.
- Time
 - The time when the message was sent or received.

Description

 This will be a pre-filled section that contains the name of the item a given message corresponds to so the landlord knows.

Content

 The body of the message, contains any questions the renter has as well as at least one form of contact i.e. phone number, or email.

-Media Storage

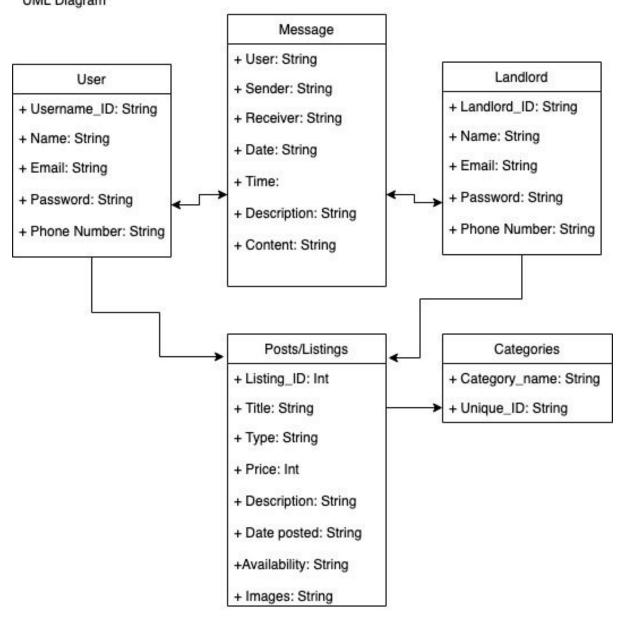
- Image format .png, .jpg, .gif
- Video format .mp3, .m4a, .wma, .wav
- Images uploaded by landlord will be stored on a file system on the server. The
 database will serve these image files through use of the stored path relating to a
 given image file. We chose this option as it seems to be a widely accepted
 standard, and reinforces good security measures.

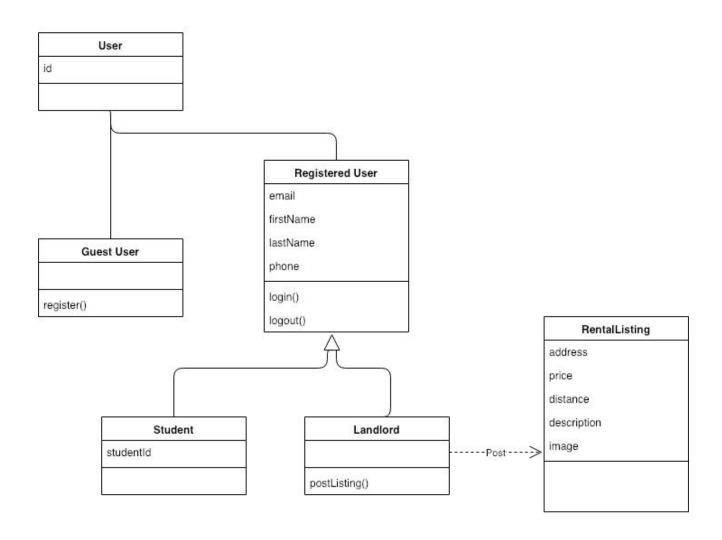
-Search/Filter Architecture and Implementation

- The approach for the architecture of search will implemented using MySQL to form our queries.
- When selecting a category with no search keywords entered, we will dump all
 posts belonging to the given category from the database and dynamically load
 the page.
- Similarly, when no category is given and only a keyword is given, we will dump the entire database (from all categories) for which we get a close match on on the keyword to any part of the title.
- In case of no results given for a particular combination of category and/or keyword, we do our 'default' dump of the database (this is what is displayed on the homepage).
- The page will display the newest posts at the top of the page and older posts will follow chronologically.

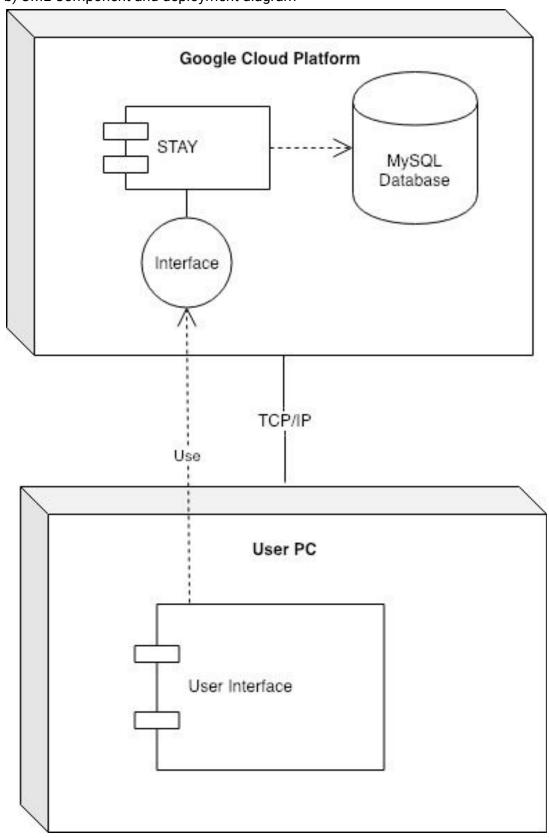
5. High Level UML Diagrams

a) High-level UML class diagrams for implementation classes of core functionality.
 UML Diagram





b) UML Component and deployment diagram



6. Identify Risks

Risks:

Skills Risks: The majority of team is unfamiliar and new to the development stack such as node.js/react/express, and need to spend time to gather and learn from a variety of online resources and tutorials.

Schedule Risks: The development team members are mostly in their senior year of university, so that there are risks that the schedules conflict and time management occur the fact that individual projects, jobs and etc.

Technical Risks: Due to the various features among the web server options, the billing risks can occur due to the testing developments such as testing with one or more instances and stacks on Web server.

Teamwork Risks: Collaboration and contribution are essential for the team and stimulating participation and assigning tasks individual may help during the development of the software stack.

Legal/Content Risks: For the demo purpose of the website, our team is going to be creating the majority of the example listings. The main risk involved is choosing to use pictures of houses with the proper permissions and copyright, so our team will use the images of public places and only for non-commercial purpose.

Resolutions:

To resolve the mentioned risks above, meeting in-person or via zoom video conference will be held periodically as needed during the development time, discussing the application, framework, new resources and learning strategies. We also use the project management tool called Trello to assign task with soft deadlines and hard deadlines. This tool in general can easily visualize, convenient and effective to use. Backend team and frontend team work collaboratively as in need and discuss the structure core standard of the implementation by helping each other. Team lead take care in need of necessary individually or as in team in effective ways.

7. Project management

In developing the STAY web application for SFSU students, project management is an essential role. To perform the effective strategies, our team decided to use the Trello project management tool.

The great features of this tool is that it is easy to use and stay well organized. Each team member can easily see the tasks, status and deadline obviously on the Trello dashboard. We assigned each milestones with clear deadline date and team lead, frontend/backend lead assign individual tasks to perform the team members. Before the deadline, team lead monitors process and performance of individual and offer additional technical help if need. The main collaboration and communication task can be done using the Slack application and if need via direct message or a phone call or video conferencing.

When there are major frameworks and technologies in need of be attention, we communicate via email with CTO Anthony and Nicholas and if needed, or visiting them during office hours and a small meetup with them to solve some technical issues that may arise.