CSC 648/848 SFSU Fall 2023 Milestone 3

Review of functionality, UI\UX, SW and planning for final product delivery

Team Member	Roles
Jeremy Tran	Team Lead / Github Master /jtran43@mail.sfsu.edu
Geovanni Valadez	Front End Lead gvaladez@mail.sfsu.edu
Mozhgan Ahsant	Scrum Master mahsant1@mail.sfsu.edu
Anthony Silva	Database Lead asilva32@mail.sfsu.edu
Aman Khera	Back End Lead akhera@mail.sfsu.edu
Ivan Ayala-Brito	Documentation Master iayalabrito@mail.sfsu.edu
Daniel Enriquez	Front End Engineer denriquez@mail.sfsu.edu
Alex Huang	Back End Engineer xhuang20@mail.sfsu.edu

History Table

Milestone Version	Date Submitted
M1V1	9/21/23
M1V2	10/12/23
M2V1	10/12/23
M2V2	11/2/23
M3V1	11/2/23
M3V2	11/30/23
M4V1	11/30/23

1. Data Definitions

1.User

- Unregistered: Limited to view-only access.
- Key Attitributes: N/A
- Relationships: None

2. Background Check

- Assesses registered user’s financial and rental history.
- Key Attributes: Check_ID, Status
- Relationships: Linked to a registered user via Check_ID.

3. Registered User

- Full site access with functionalities like posting, messaging and joining groups.
- Key Attributes: User_ID, Password, Check_ID
- Relationships: Can have multiple rental listings, posts, messages, notification, events and linked to a background check.

4. Location of Rental Listings

- Precise location for rental listings.
- Key Attributes: Location_ID, Listing_ID
- Relationships: Linked to Rental Listings via Listing_ID.

5. Region

- General location of the rental listing such as neighborhood.
- Key Attributes:Region_ID, Location_ID
- Relationships: Contains multiple Locations

6. Rental Listings

- A post that contains media, images or videos along with information regarding the property.
- Key Attributes: Listing_ID, User_ID
- Relationship: Belongs to a registered user.

7. Posts

- Generic posts with or without media.
- Key Attributes: Post_ID, User_ID, Comment_ID
- Relationships: Owned by a registered user and can have multiple Comments.

8. Messages

- Private communication between registered users.
- Key Attributes: Message_ID, User_ID(to), User_ID(from)
- Relationships: Between two registered users.

9. Notification

- Updates for users about thei account activities.
- Key attributes: Notification_ID, Notification_Type
- Relationships: Received by Registered User.

10. Events

- User-created events either open or by invitation.
- Key attributes: Event_ID, User_ID(Host)
- Relationships: Hosted by a Registered User and contains other Registered Users as event guests.

11. Friend Request

- Registered users can add other registered users to their friend list.
- Key Attributes: Request_ID, User_ID(to), User_ID(from)
- Relationships: Between two Registered Users.

12. Saved Properties List

- Registered users can save preferred rental listings.
- Key Attributes: List_ID, Saved_ID
- Relationships: Owned by a Registered User.

13. Housing Cost Calculator

- Provides rental/buying suggestions based on input.
- Key Attributes: N/A
- Relationships: None(data is not saved)

14. Groups

- For group communications and activities.
- Key Attributes: Group_ID, User_ID

- Relationships: Owned by a Registered User and can contain other Register Users.

15. Reports

- Allows for users to report rental listings.
- Key Attributes: Report_Id, User_ID
- Relationships: Linked to a Rental Listing.

2. Functional Requirements

Priority 1:

1. **User:**

- (1.1) Users shall be able to register an account using a unique email and set up a password.

- (1.2) Users shall be able to browse the site without an account but will not be able to interact with posts or other users.

- (1.3) A user can reset their password if forgotten.

- (1.4) The system should verify user identities securely through email or phone number verification.

2. **Background Check Integration**

- (1.5) All registered users that wish to rent or rent out their place shall be subject to a background check.

- (1.6) The system shall integrate with predefined screening criteria and business rules, allowing property managers to set specific eligibility criteria based on factors such as credit score, criminal history, rental history, and income.

- (1.7) Registered users can open the status of the background check.

3. **Registered User**

- (1.8) Registered users shall be given the option to make posts on the site.

- (1.9) A Registered User shall be able to link their social media accounts such as Instagram and Twitter.

- A Registered User shall be able to apply filters when searching for rental listings

- A Registered User shall be able to search specifically for number of bedrooms, bathrooms, price, and apartment type.

- A Registered User shall be able to update their profile based on preferences they want displayed

4. **Location of Rental Listing**

- (1.10) Locations shall include campus area, neighborhood, and proximity to public transport.

- (1.11) Users should be able to filter and view listings by clicking on map markers or drawing custom search areas.

- (1.12) Rental listings shall be separated into regions, where users can view by clicking interactable map region.

5. **Rental Listing:**

- (1.13) Landlord accounts shall be able to create, update, or delete their rental listings, specifying details such as the number of bedrooms, bathrooms, etc.

- (1.14) Landlord accounts shall be given the permission to add rental listing to the site.

6. **Housing Cost Calculator:**

- (1.15) Registered Users can generate approximate costs for chosen intervals and forecasts for the property.

7. **Posts:**

- (1.16) A post shall be able to contain text
- A post shall be able to contain photos
- A post shall be able to contain links
- A post shall be able to contain user tags
- A post shall be able to contain likes

- (1.17) Posts shall allow comments, each with a unique commentID and creation timestamp.

- A comment shall be allowed to have likes

- (1.18) Posts shall be only edited by the author.
- (1.19) Posts with comments shall only be edited by the author.

Priority 2:

1. **Registered User:**

- (2.1) Registered users shall save rental listings to their saved properties list.

- (2.2) A Registered User shall be able to form groups of other registered users they wish to house with.

2. **Rental Listing:**

- (2.3) Rental Listing shall have a minimum amount of information and media needed for posting.

3. **Messages:**

- (2.4) Registered users shall be able to initiate private messages with one other registered user.

4. **Saved Properties List:**

- (2.5) Users can create lists of saved properties.

- (2.6) Each saved list shall have a unique ListID.

- (2.7) Properties List shall be under a registered user and not viewable to other non-registered or registered users.

- (2.8) Properties list shall group properties by similarities or by a key word as desired by a registered user.

5. **Groups:**

- (2.9) A group shall share a saved properties list and all receive notifications for a property.

6. **Reports:**

- (2.10) Registered users can generate reports on rental listings, user activity, and feedback.

7. **Housing Cost Calculator:**

- (2.11) The Calculator shall generate a curated list of houses best fit for a registered user based on filters and location.

8. **Reports:**

- (2.12) The system shall provide functionality to generate various types of rental reports, including rental transaction summaries, rental history for specific customers, and rental equipment utilization reports.

- (2.13) Users shall be able to customize the content and format of reports, including selecting specific date ranges, sorting options, and the inclusion/exclusion of specific rental data field, such as rental duration, customer information, and equipment details.

Priority 3:

1. **Registered User:**

- (3.1) Registered users shall send and receive friend requests.

2. **Location of Rental Listing:**

- (3.2) Users may opt to receive notifications when new rental listings by distance from a specified location (e.g., within 5 miles of a particular address or point of interest).

- (3.3) The system shall provide autocomplete suggestions to assist users in specifying the location.

3. **Posts:**

- (3.4) A post shall be able to get pinned by the creator with a maximum of three for his or her account.

- (3.5) A post shall be able to get promoted by the creator at least twice with the boost lasting two days.

4. **Messages:**

- (3.6) The system should allow users to organize their messages, including archiving or deleting old messages.

- (3.7) Users should receive real-time notifications (e.g., push notifications or email notifications) when they receive a new message or when there are updates to existing messages.

5. **Notification:**

- (3.8) Registered users shall receive a notification upon receiving a new message.

- (3.9) A notification shall be able to be turned off or on by a register user or group.

6. **Events:**

- (3.10) Registered users shall be able to set up events (open house, etc) for other users to see.

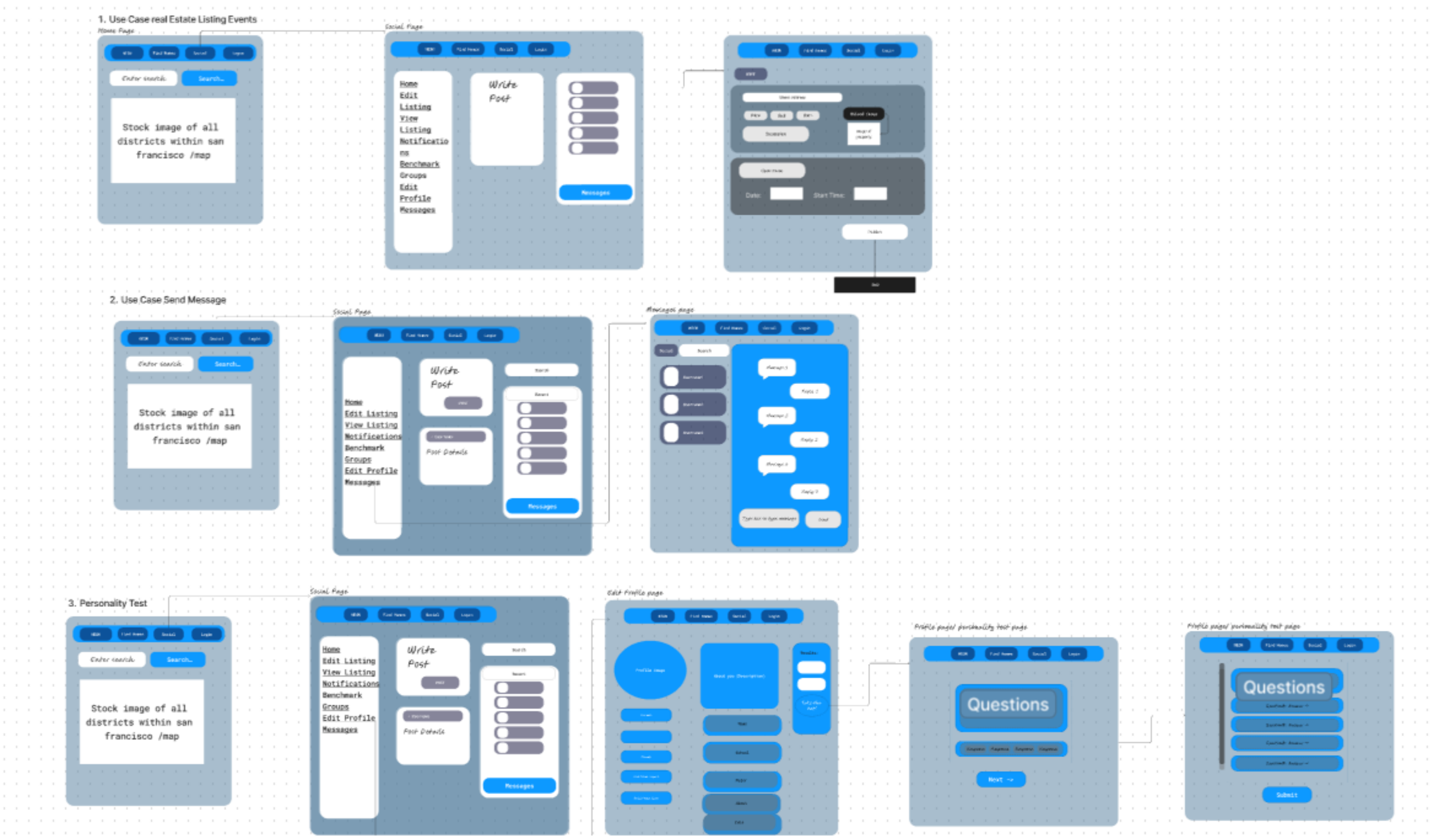
- (3.11) An event shall have its own location, date, eventID, and a list of confirmed attendees.

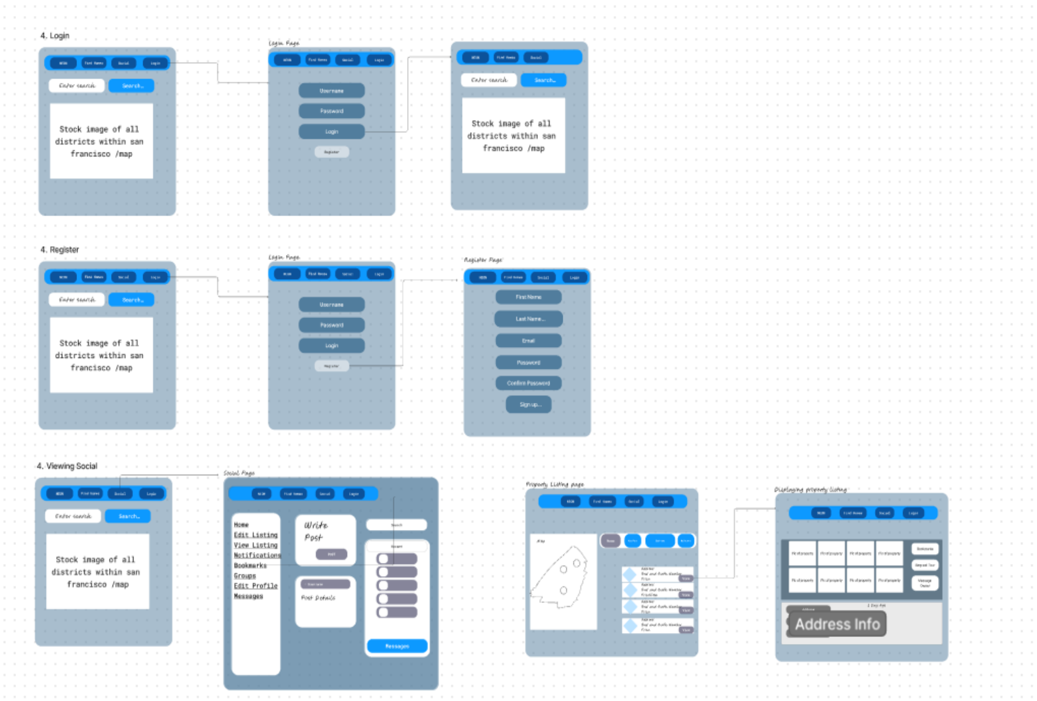
- (3.12) An event shall only be made by verified registered users and have an expected list of attendees.

7. **Friend Request:**

- (3.13) Register users sending the friend request shall allow permission changes for the requesting friend to view content

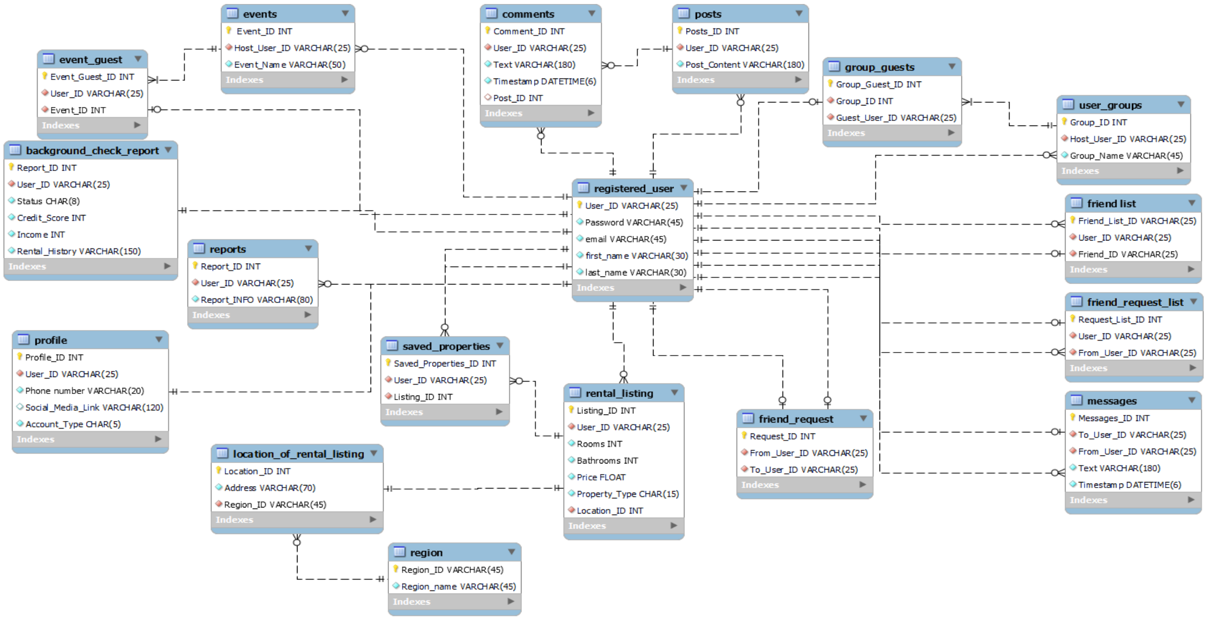
3. Wireframes Based on your Mockups/Storyboards



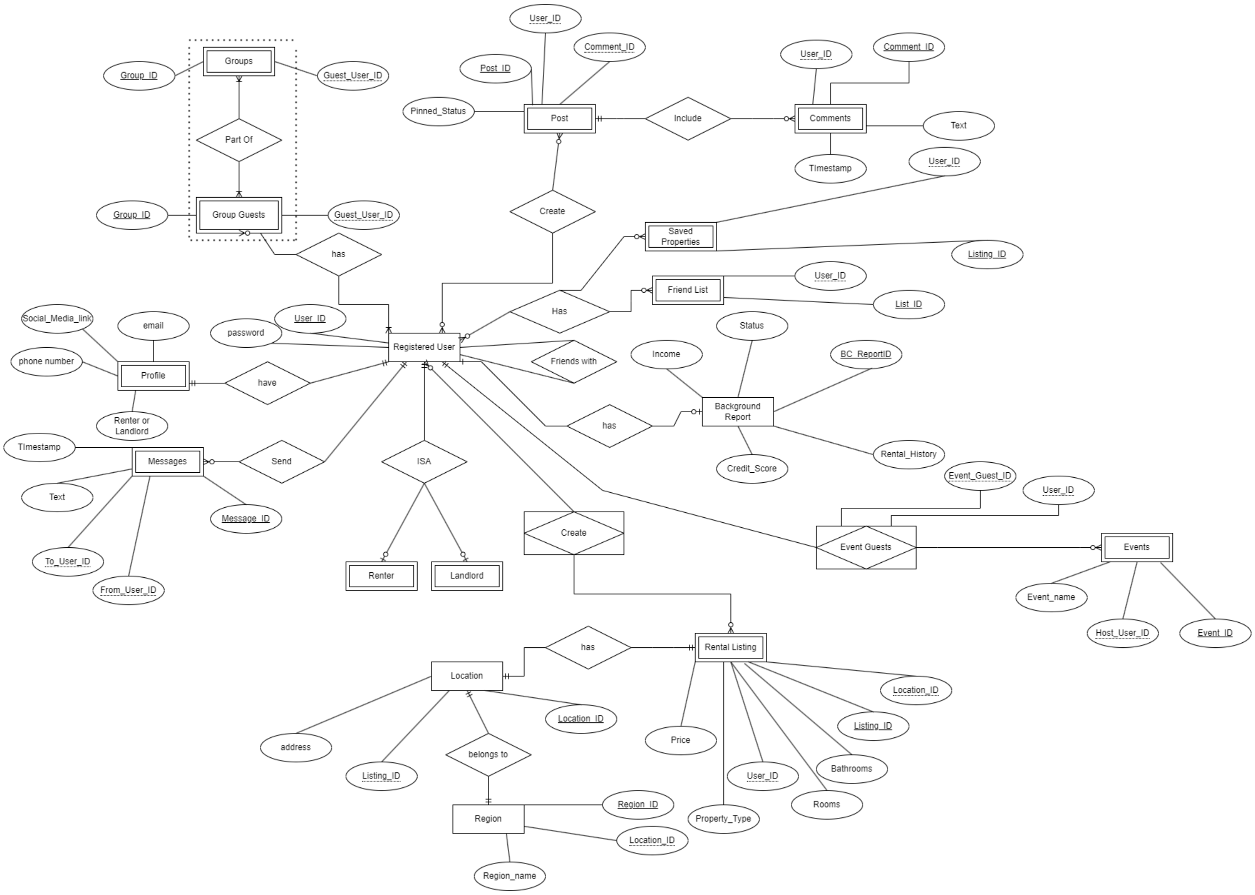


4. High Level Database Architecture and Organization:

EER:

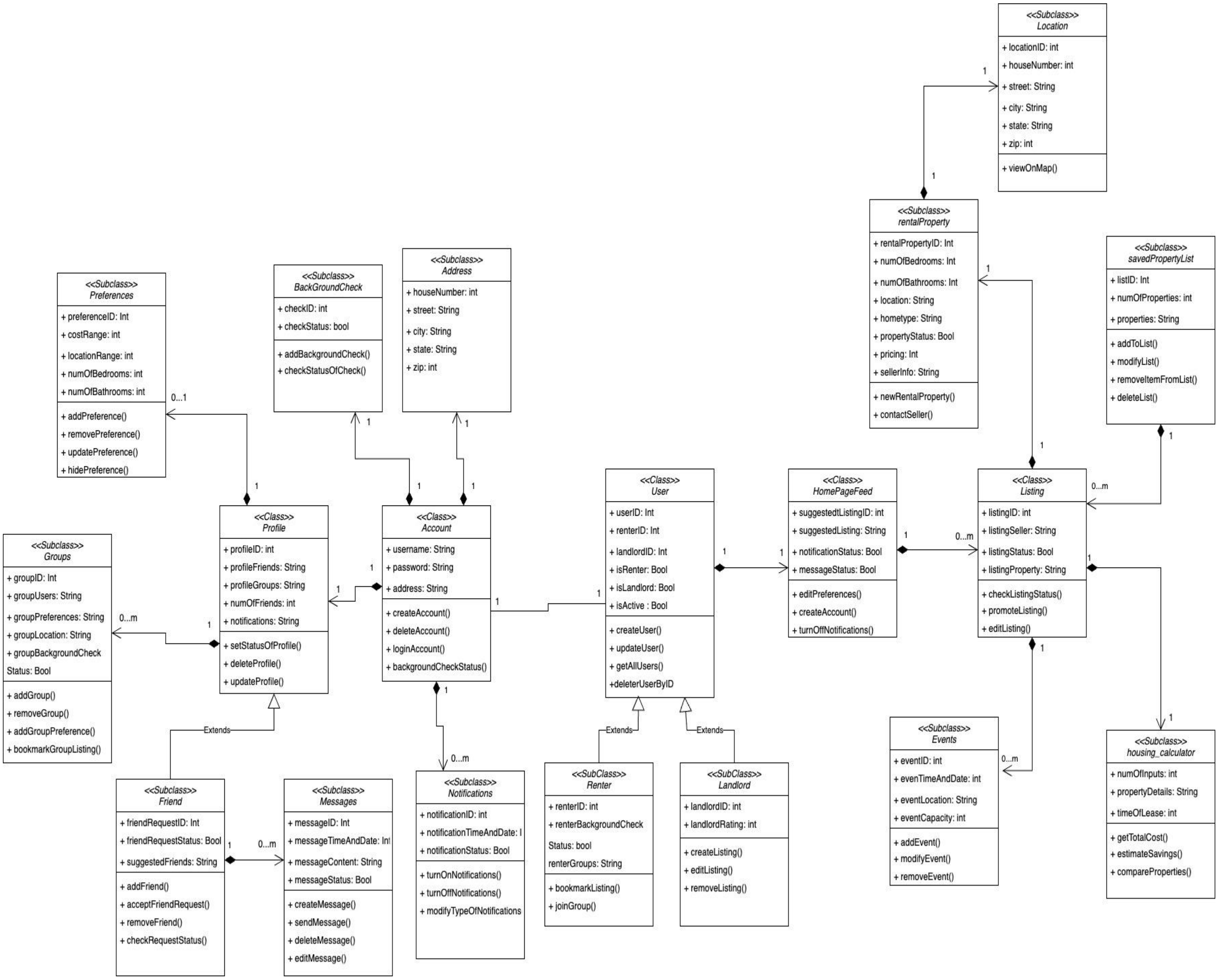


ERD:

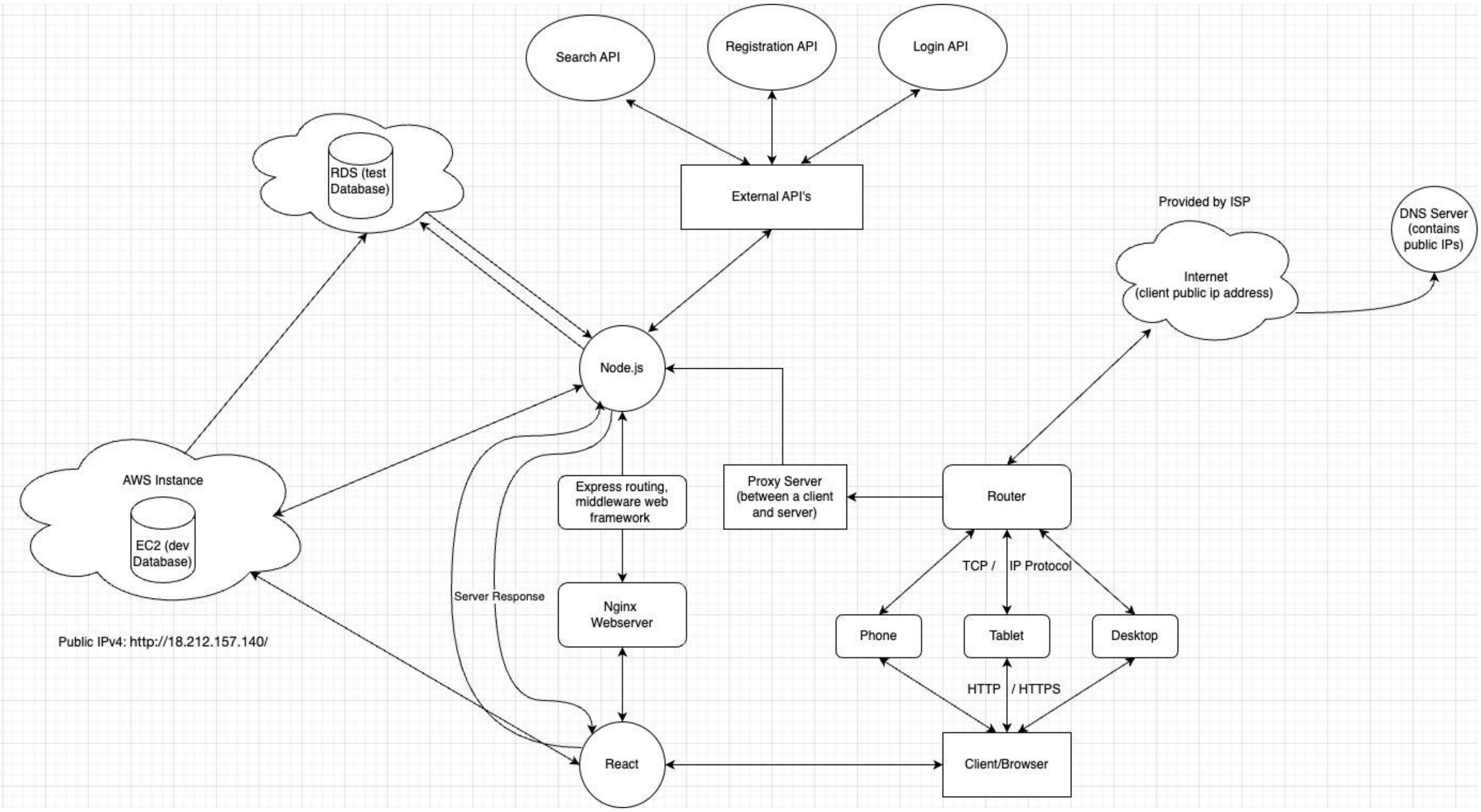


5. High Level Diagrams

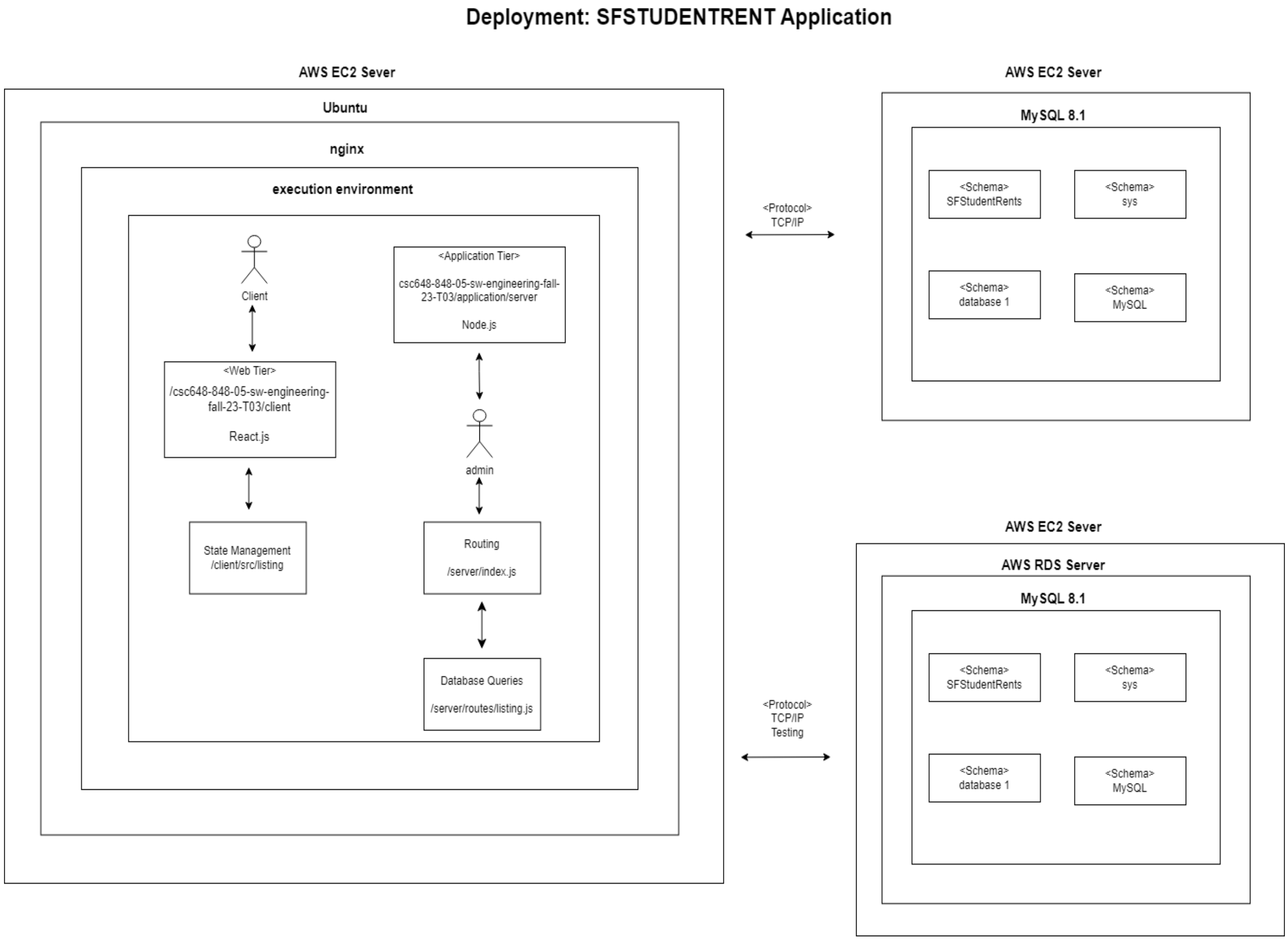
UML Diagram:



Application Network Diagram:



Deployment Diagram:



6. List of Contributions in this milestone

Members	Contributions	Score
Jeremy Tran	● Completed the Registration function for backend and integrated it with the frontend UI component ● Assisted the backend team on connection between backend and database ● Organized and managed all weekly meetings ● Reviewed all work within 1 day according to due dates set by the frontend, backend and database ● Did the M2 revision for storyboard and data definitions	
Geovanni Valadez	● Organized the frontend team and the meetings ● Completed multiple pages and did the UI across the different pages ● Organized the navigation between all pages ● Took on additional responsibilities to help complete the frontend ● Did M2 revisions for section 3	9
Mozhgan Ahsant	● Attended every weekly group meeting and all the subteam meeting ● Attempted the login function	6
Anthony Silva	● Attended every group meeting and was an active participant ● Completed the EER ● Did the database schema and did all revisions needed for the databse	8
Aman Khera	● Organized the backend team and set up project timeline ● Completed the search algorithm, crud functions and login ● Integrated the backend with database and frontend ● Deployed the final product	10

	● Sorted out Git conflicts ● Attended all team meetings and organized backend team meetings	
Ivan Ayala-Brito	● Worked on the frontend pages ● Attempted the storyboard ● Organized the documents for M2V2 and M3V1	7
Daniel Enriqurez	● Set up the Trello for frontend ● Worked on the Figma wirefram	3
Alex Huang	● Worked on and created the crud functions ● Did the search algorithm ● Finished the deployment diagram ● Attending all team and subteam meetings	8