CS 491: Senior Design Project

Analysis Report

BilMate : Group T2335

Ahmet Salman	21901004
Atasagun Samed Şanap	21902435
Ebrar Bozkurt	21802824
Javid Moradi	21903645
Onuralp Avcı	21902364

13/11/2022

Jury Members: Erhan Dolak, Tağmaç Topal, Uğur Doğrusöz

Innovation Expert: Ahmet Kocamaz

1.0 Introduction

Nowadays, especially in Turkey, one of the leading and most crucial problems university students face is finding a proper place to live while studying for a degree. Although there are existing student dormitories for students, there are times when they are full, in bad condition, expensive, or very far away from the university of students. Including the motivation to live in an individual room rather than a shared room with 2, 3, 5, or even 8 people, there are many reasons for students to share a house with other students, preferably with the students of their university. Yet this can be tiring and challenging as there is no easy way of finding a place at a good location, shared with trusted students, for a preferred rent, etc. Some university confession pages, such as “Bilkent Itiraf” currently act as a meeting point for people looking for a housemate and those wanting to become a housemate. Yet, as those platforms are not specialized for such issues, they are very inefficient in achieving suitable matches that satisfy both sides.

At this point arises the need for a platform where students looking for housemates and students who want to become one can meet and browse through the options based on their price range, house location, number of housemates, ownership of pets, habits like smoking, etc. This platform will be the mobile application intended to be created through the specifications on this report, where users will register, enter their preferences and details and look for their best matches to share a house with.

2.0 Current System

There currently does not exist a system that is specifically tailored for university students. There are accommodation-sharing websites. However, they are severely lacking both in quantity and quality. For instance, the website expat.com[1] has a total of four adverts showing rooms, which is not nearly enough for students. Another website is erasmusu.com[2], but it does not offer much flexibility in requirements. Moreover, it includes a very simple filtering system, and everything is on one page. So it does not include any matching, advanced filtering, or any quality-of-life improving features

[1] https://www.expat.com/en/housing/middle-east/turkey/flat-share-house-share.html
[2] https://erasmusu.com/en/erasmus-istanbul/roommate

3.0 Proposed System

3.1 Overview

The solution to solve the shared accommodation problem mentioned above will be provided through a social platform as a mobile application for the students to use. The app will provide two options to newcomers, who can either be students looking for a housemate to share the rent and other costs with or those already looking for a house that other students are sharing.

At this point, both sides will start this process of finding housemates with their criteria. These criteria will reflect their priorities, such as the location of the house in terms of security, the proximity to the university or transportation lines, rent and other expenses, etc., as well as their concerns or preferences, such as pet allowance, smoking or drinking rules, loudness and socialness within the house, cleanliness, etc. All of these criteria and more will need to be filled in by the users. There is supposed to be a large number of criteria. Although only the crucial ones will be asked in the registration step of the application, the other less important ones, such as the preferred departments of housemates or the number of housemates, will be asked at intervals during the use of the application, for the convenience of the use of the application.

With the data retrieved through the questionnaire, the app will find matches between the housemate-seekers and house-seekers. It will be frequently pointed out in the application that with more data fields that the user fills, it will be more probable to find a good match between the two user types. Having found a good match that satisfies the users, they will be able to contact each other either through the messaging portal of the application or through their contact details, such as phone numbers or e-mail addresses, in case they want to share. It will also be possible with the app to browse through listings of houses or housemates rather than go for a matching process.

BilMate will bring sustained innovation into the market of house-sharing in Turkey in the form of a service that will transform the way students choose their roommates or find a proper place to live in. Rather than going through 10s or even 100s of people trying to find a suitable room and roommate combination, BilMate will optimize that process by allowing the user to decide for himself/herself what a suitable roommate should be.

For the initial implementation, we are planning on prioritizing the essential features that allow the user to list posting, create a posting, edit their profile, etc. During the later stages of development, we would start implementing and refining the matching algorithm to suit each

user's needs. The most substantial risk that we might face during the development stage is the accuracy and reliability of the matching algorithm. If the algorithm does not prove to be sufficient on its own, we will add extra manual filtering and sorting to accompany it in order to offer a complete experience to the user. However, if the algorithm proved to be sufficient with respect to the criteria that were established, a manual filtering system may not be needed but may still be implemented for convenience purposes.

3.2 Functional Requirements

3.2.1 Users

3.2.1.1 General Requirements

- Unregistered users should be able to register an account using their verified email.
- Unregistered users should be able to view the postings but without seeing any contact information
- Unregistered users should be able to verify their emails
- Registered users should be able to log into their accounts.
- Registered users should be able to log out of their accounts.
- Registered users should be able to change their password.
- Registered users should be able to modify their profile.
- Registered users should be able to change their privacy settings to show/hide their phone number

3.2.1.2 House-Seeker Requirements

- House-seeker should be able to answer questions indicating the preferred traits of their roommate.
- House-seeker should be able to answer any number of questions to refine the matching algorithm.
- House-seeker should be able to stop answering questions at any time.
- House-seeker should be able to filter and sort through post listings manually.
- House-seeker should be able to add specific posts to a Favorites list.
- House-seeker should be able to remove posts from their Favorites list.

3.2.1.3 House-Sharer Requirements

- House-sharer should be able to create their own posts to indicate that they are looking for a roommate.
- House-sharer should be able to add contact information to their posting.
- House-sharer should be able to modify or remove their contact information from their posting.
- House-sharer should be able to delete their own posts.

- House-sharer should be able to indicate their preference on the post before it is published.
- House-sharer should be able to edit their posts once they are published.
- House-sharer should be able to report other users in case the other party breaks the TOS.

3.2.2 Admin

- Admins should be able to ban/unban users from the platform.
- Admins should be able to review report cases.
- Admins should be able to provide a verdict regarding reported users.

3.3 Nonfunctional Requirements

3.3.1 Usability

It is essential to hide the complexity behind the UI layer in apps with complex algorithms. Regardless of what is going on in the domain layer, users should not be bothered with that part, and a seamless flow should be offered to them. In BilMate, there will be many questions for users in order to make the matching algorithm as accurate as possible. However, the app will be optimized to make that questioning flow as user-friendly as possible.

3.3.2 Security

All information shared by the users can be considered sensitive data. This means that on both the backend and frontend sides, BilMate must be developed with a “data protection by design and default” strategy. All the sensitive content will be encrypted with a well-chosen encryption algorithm, and any data that is not required to be stored will not be stored.

3.3.3 Scalability

The app must be scalable first of all for the increasing number of Bilkent students; then for the other universities and their students as well.

3.3.4 Maintainability

User traffic on the app will fluctuate based on the time of the year, which means, especially at the start of semesters, the app will face intense user traffic. At those times, there will be some problems on the backend side due to high user requests, and possibly some UI bugs will appear on the frontend side. The architecture must allow developers to fix problems and maintain the app as fast as possible for those scenarios.

3.4 Pseudorequirements

- Kotlin will be used to develop the frontend

- FastAPI will be used to develop the backend
- MongoDB will be used as the database
- The application will be an android OS mobile application

3.5 System Models

3.5.1 Use Case Model

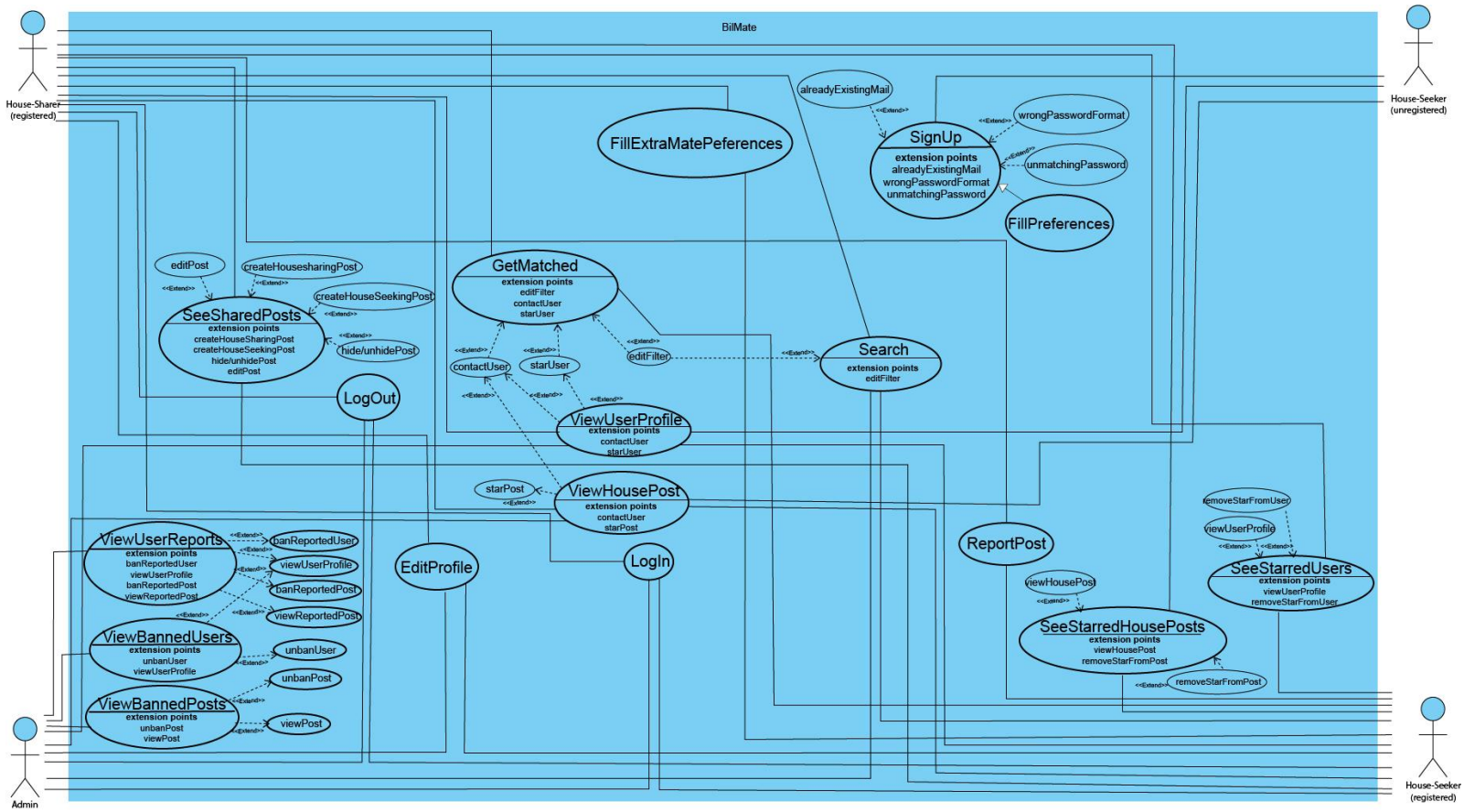


Figure 1: Use case diagram

3.5.2 Scenarios

Scenario 1

Use Case Name: SignUp
Actors: House-Seeker (unregistered)
Entry Conditions: House-Seeker is unregistered
Exit Conditions: House-Seeker is registered
Flow of Events:

- Goes to signup page

- Enters email and password
- Fills up mandatory user preferences and details
- User is registered

Scenario 2

Use Case Name: LogIn
Actors: Admin, House-Sharer and House-Seeker (registered)
Entry Conditions: actor is registered
Exit Conditions: actor is logged in
Flow of Events:

- goes to login page
- Enters credentials
- Logs in

Scenario 3

Use Case Name: SeeSharedPosts
Actors: House-Sharer, House-Seeker
Entry Conditions: Actor is logged in
Exit Conditions: Actor is logged in and has seen his/her shared houses or shared profile posts
Flow of Events:

- Actor clicks on “HouseSharing” posts or “HouseSeeking” posts
- Actor views posts
- Actor can, hide, unhide, edit or create posts
- Actor leaves

Scenario 4

Use Case Name: GetMatched
Actors: House-Seeker, House Sharer
Entry Conditions: Actors are logged in
Exit Conditions: Actors have a list of matchs
Flow of Events:

- Actor goes to match page
- Actor is shown matches
- Actor can edit filter for better results
- Actor can star user or star post
- Actor can contact user
- Actor can exit

Scenario 5

Use Case Name: FillExtraMatePreferences
Actors: House-Sharer, House-Seeker
Entry Conditions: Actors are logged in

Exit Conditions: Actors will have more preferences
Flow of Events:

- Actor goes to profile
- Actor fills up extra preferences
- Actor saves and closes

Scenario 6

Use Case Name: Search
Actors: House-Sharer, House-Seeker, Admin
Entry Conditions: Actors are logged in
Exit Conditions: Actor will have a list of related search
Flow of Events:

- Actor goes to search field
- Actor input search parameters
- Actor filters out the search

Scenario 7

Use Case Name: SeeStarredUsers
Actors: House-Seeker, House-Sharer
Entry Conditions: Actor is logged in
Exit Conditions: Actor will have seen the starred users
Flow of Events:

- Actor goes to starred users
- Actor sees a list of previously starred users
- Actor can view user profile
- Actor leaves

Scenario 8

Use Case Name: SeeStarredHousePosts
Actors: House-Seeker, House-Sharer
Entry Conditions: Actor is logged in
Exit Conditions: Actor will have seen previously starred shared house posts
Flow of Events:

- Actor goes to starred house posts
- Actor sees a list of previously starred houses
- Actor can view house posts
- Actor leaves

Scenario 9

Use Case Name: Report Post
Actors: House-Seeker, House-Sharer
Entry Conditions: User is logged in, and viewing a post

Exit Conditions: User is reported
Flow of Events:

- User is viewing a house post
- User reports the post

Scenario 10

Use Case Name: ViewUserReports
Actors: Admin
Entry Conditions: Actor is logged in
Exit Conditions: Actor will view user reports and bans users or posts
Flow of Events:

- Actor views user' reports
- Actor reviews them individually
- Actor can view reported user profile
- Actor can view reported post
- Actor can choose to ban the reported user
- Actor can choose to ban the reported post
- Actor leaves

Scenario 11

Use Case Name: ViewBannedUsers
Actors: Admin
Entry Conditions: Actor is logged in
Exit Conditions: Views banned users and decides to unban them or not
Flow of Events:

- Actor views banned users
- Actor can view the banned user profile
- Actor can choose to unban the user
- Actor leaves

Scenario 12

Use Case Name: ViewBannedPosts
Actors: Admin
Entry Conditions: Actor is logged in
Exit Conditions: Views banned posts and decides to unban them or not
Flow of Events:

- Actor views banned posts
- Actor can view a post
- Actor can unban the post
- Actor leaves

Scenario 13

Use Case Name: EditProfile
Actors: Admin, House-Sharer, House-Seeker
Entry Conditions: Actor is logged in
Exit Conditions: Actor edited profile
Flow of Events:

- Actor goes to profile
- Actor edits profile
- Actor saves and exits

Scenario 14

Use Case Name: LogOut
Actors: Admin, House-Sharer, House-Seeker
Entry Conditions: Actor is logged in
Exit Conditions: Actor logged out
Flow of Events:

- Actor goes to profile
- Actor logs out

3.5.3 Object Model

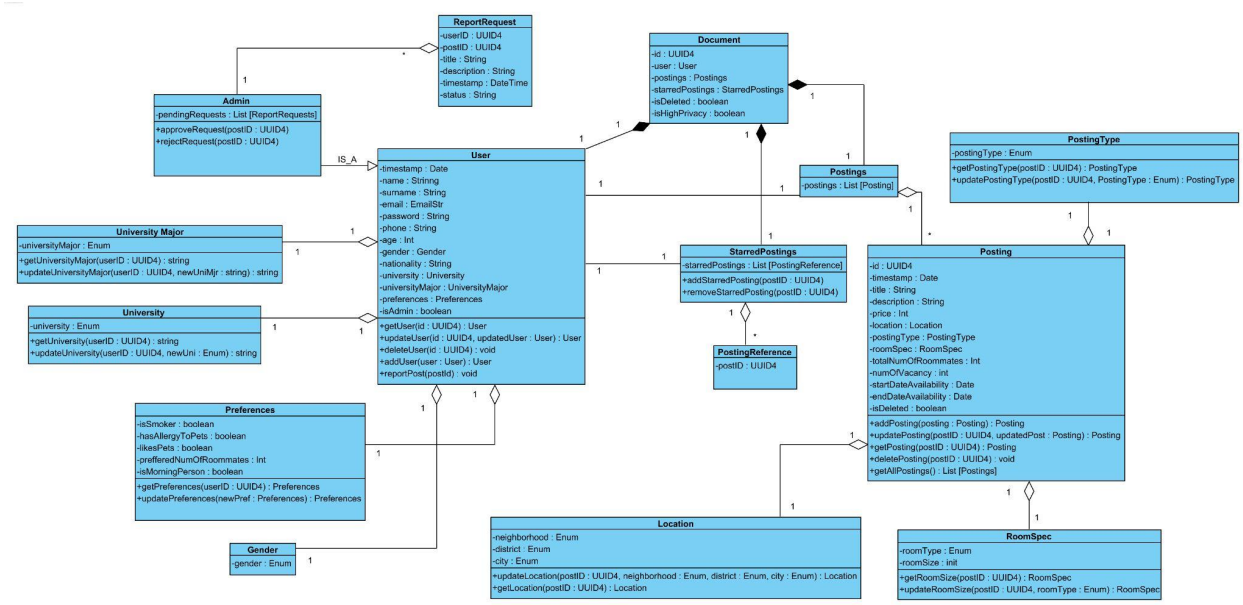


Figure 2: Object / Class diagram

Class	Explanation
Document	Document is the class responsible for “gluing” together all the functions needed, it provides a high-level abstraction of all the other classes
User	The User class for all the functions and properties belonging to a user including the creation and modification of the user
Postings	This class hold a list of Posting objects which in turn is held by the Document class
Posting	This class contains all the functions and properties for the Posting object, which a User can have many of
StarredPostings	This class holds a list of PostingReference which can be used to refer to any Posting object in the system
postingReference	This class is responsible for holding an ID referring to a Posting object
Admin	Thsi class is responsible for managing all the functions of admins which include the accepting of report requests, which results in banning
Location	This class holds an enumeration of the location for each Posting object
RoomSpec	This class holds the specifications of each room in Posting object
PostingType	This class holds an Enumeration of the posting type, whether it is a house-sharer post or a house-seeker
UniversityMajor	This class holds an enumeration of general university majors (e.g. Natural sciences, engineering, etc.)
University	This class holds an enumeration of all universities in Ankara
Preferences	This class holds the properties that will be used to create a better match for the user
Gender	This class holds an enumeration of genders

Admin	This class is responsible for all the functionalities of an admin
ReportRequest	This class holds the model for the request sent when a user reports a post

3.5.4 Dynamic Model

3.5.4.1 Sequence Diagram

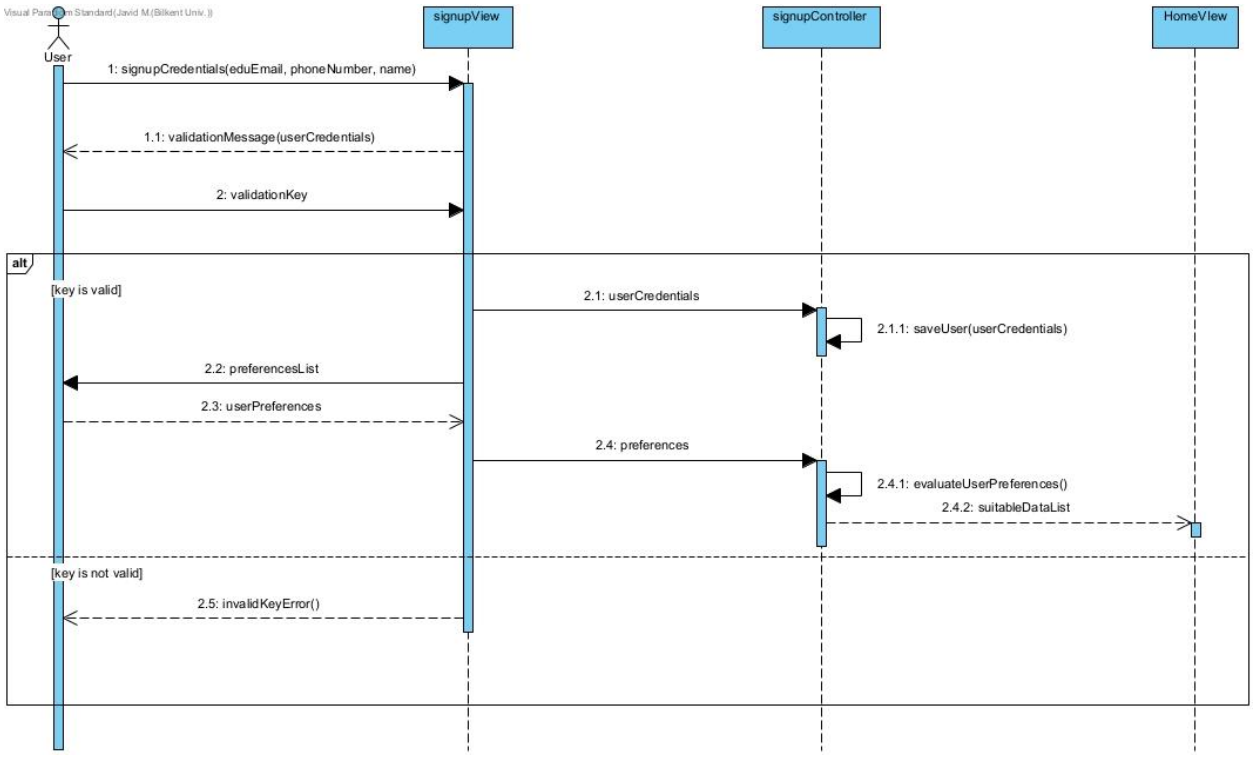


Figure 3: Sequence diagram for “signup” flow

First scenario of a sequence diagram is a new user’s sign-up/profile creation. First, a user must provide some credentials to the view, which are the user’s educational email address, phone number, and name; please note that these are noteworthy credentials, as the application might require further user credentials. Later, the application will send a verification key via email to the user and expects the user to provide it for security purposes. If the key is invalid, the user will be notified with an error message. If the key is correct, the user will be saved to the system. Afterward, the user will be given the option to answer several preference-related questions. These initial preferences will be evaluated, and suitable data will appear on the user’s homepage view.

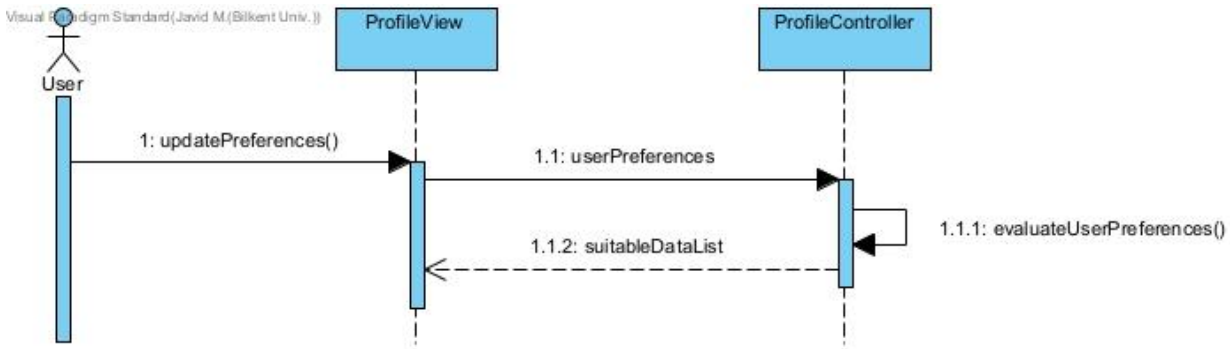


Figure 4: Sequence diagram for “update preference” flow

The second diagram is about a user updating their preferences list. In their profile section of the application, users can change their preferences, such as if they smoke, like pets, etc. The updated list of preferences will be sent to the controller, where suitable recommended data of the user will be updated, as the recommendation algorithm will update itself based on the provided new preferences list.

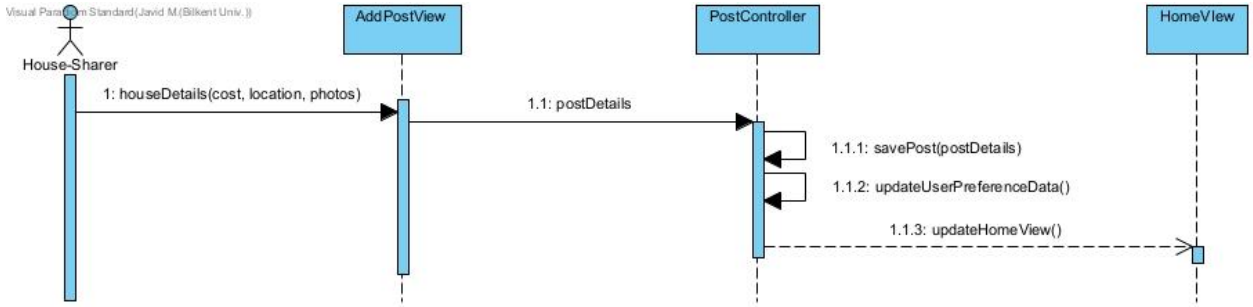


Figure 5: Sequence diagram for “share house post” flow

The diagram above is the event of a house-sharer sharing a post. First, a house-sharer will provide necessary information about their post, such as cost, location, etc. Later, these details will be sent to the system and saved. Users’ preference list will be updated according to this new post’s content, which consequently updates users’ home views. Please note that a house-sharer is also a user.

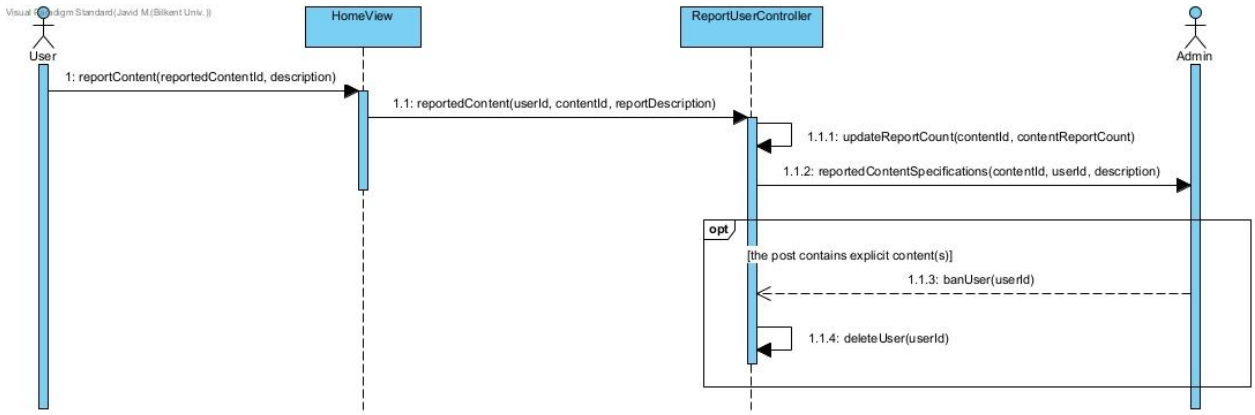


Figure 6: Sequence diagram for “report post” flow

Here, an event of a user reporting a post is displayed. First, a user reports a post by providing the reasoning behind their action. Later, the post and its other details, such as its owner, are sent to the system. Note that on each report, the post’s report counter will increase since a post can be reported by multiple users. Afterward, the reported content will be sent to an admin. Admin will examine the post with given comments, and if the decision is positive, that is, the post contains explicit material(s), the post is deleted, and the post’s owner is expelled from the application.

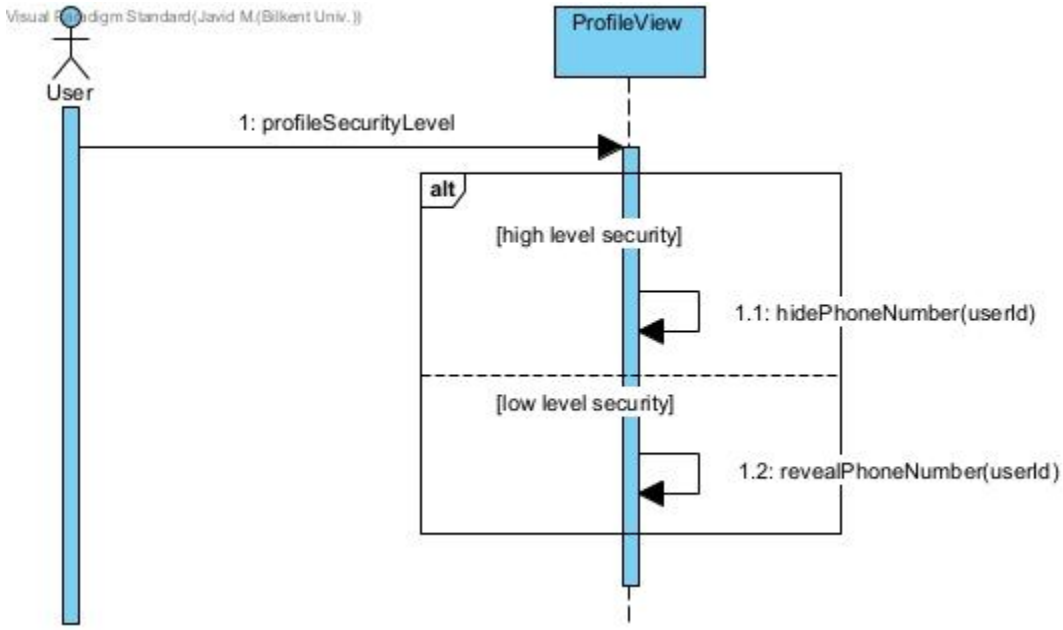


Figure 7: Sequence diagram for “set privacy level” flow

The final diagram is about users’ security regarding their phone number’s public availability. Users can either hide their phone numbers due to confidentiality-related reasons. If they do, the application will show their email address as the main communication method. If they decide to reveal their phone number, other users can contact them via email or phone call.

3.5.4.2 Activity Diagram

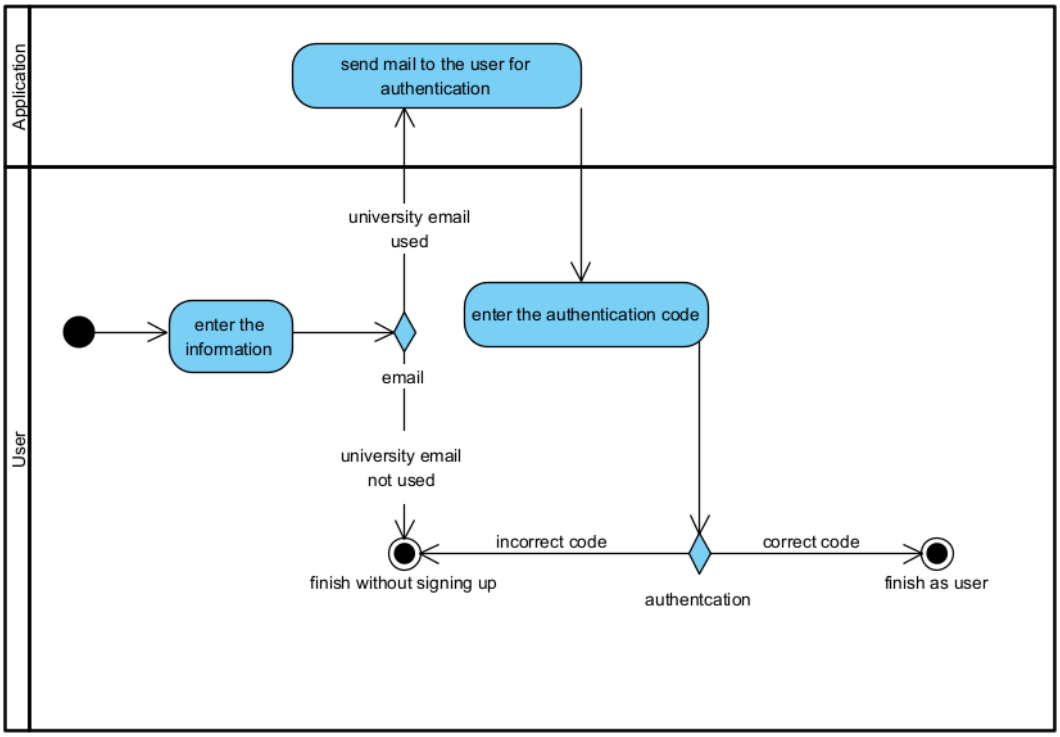


Figure 8: Activity diagram for “2FA” flow

First activity diagram shows two way authentication of the system. User first enters their information. Only university emails are accepted, otherwise the signup becomes unsuccessful. If the user uses university email, an authentication mail is sent. If the user enters the code correctly, signup becomes successful.

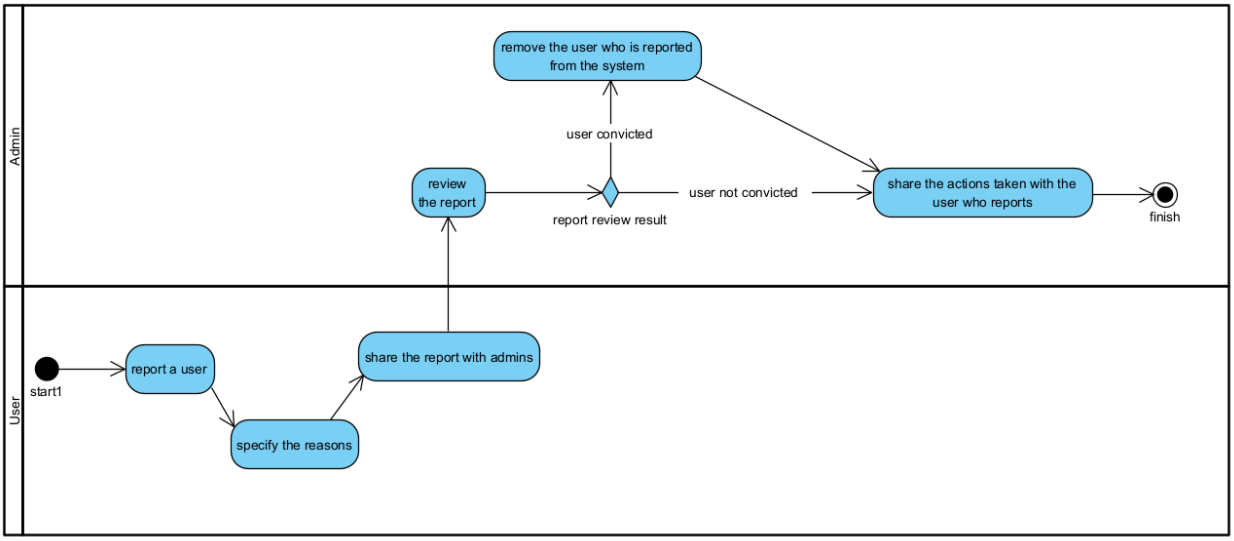


Figure 9: Activity diagram for “report user” flow

Second activity diagram shows the flow of events in reporting a user. A user can report a user by specifying the reasons. Then s/he shares the reports with admins. Admins review the report. They may either remove the user or not find the user not guilty. In both cases, the user reporting is informed about the decision.

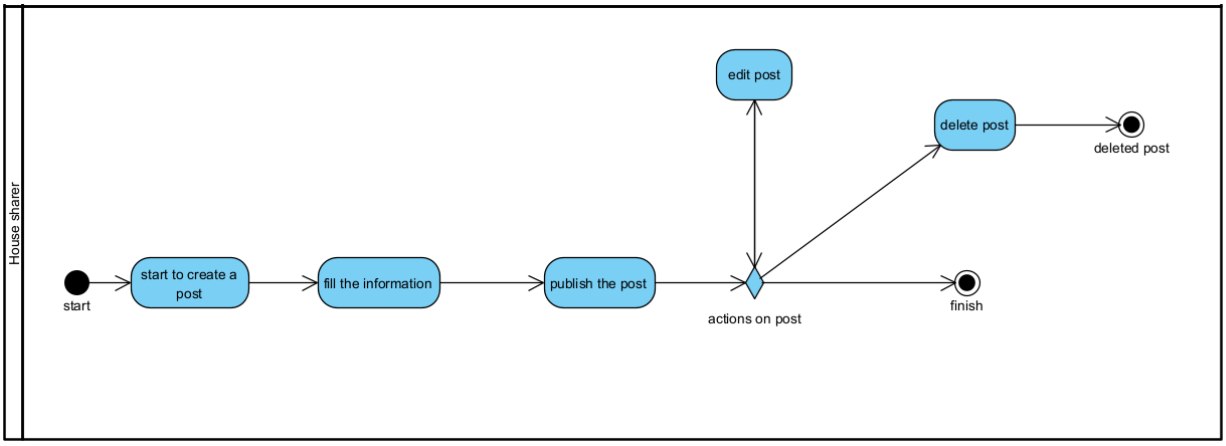


Figure 10: Activity diagram for “create post” flow

Third activity diagram shows the flow of events in creating a post. First user starts creating a post action by pressing the necessary buttons. Then, the user fills in the necessary information such as title, description, location. Then, the user can publish the post or delete the post before publishing. If s/he publishes the post, s/he can do nothing, edit the post later or delete it.

3.5.4.3 State Diagram

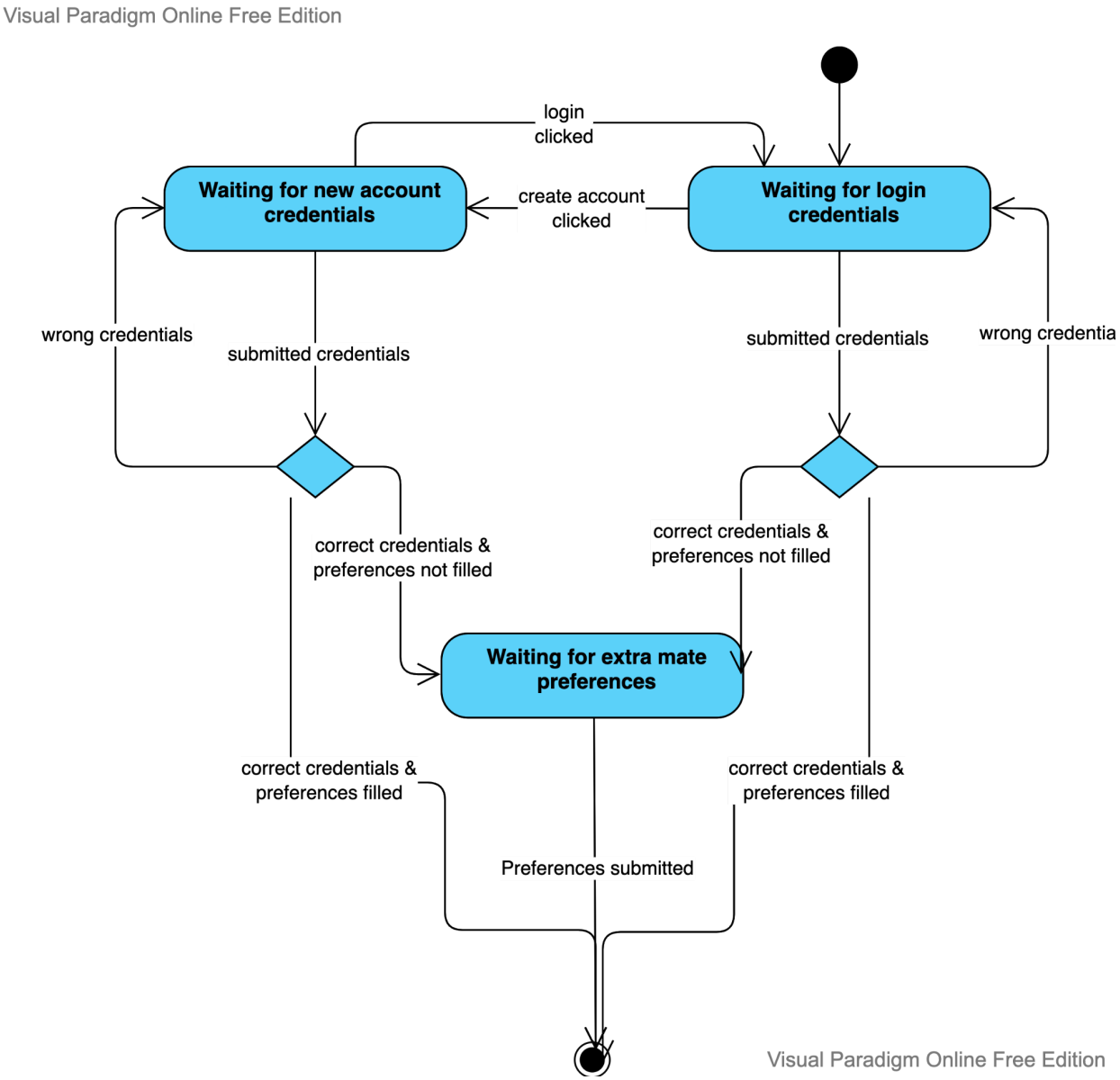


Figure 11: State diagram for "login/signup" flow

The first state diagram shows the states during login/signup flow. The initial state for this flow is waiting for login credentials. If the user is not signed up yet, he can be navigated to the signup flow. After successful login, depending on whether user preferences are filled or not, they will be navigated to either the home screen (exit state) or they will be first asked to fill their preferences.

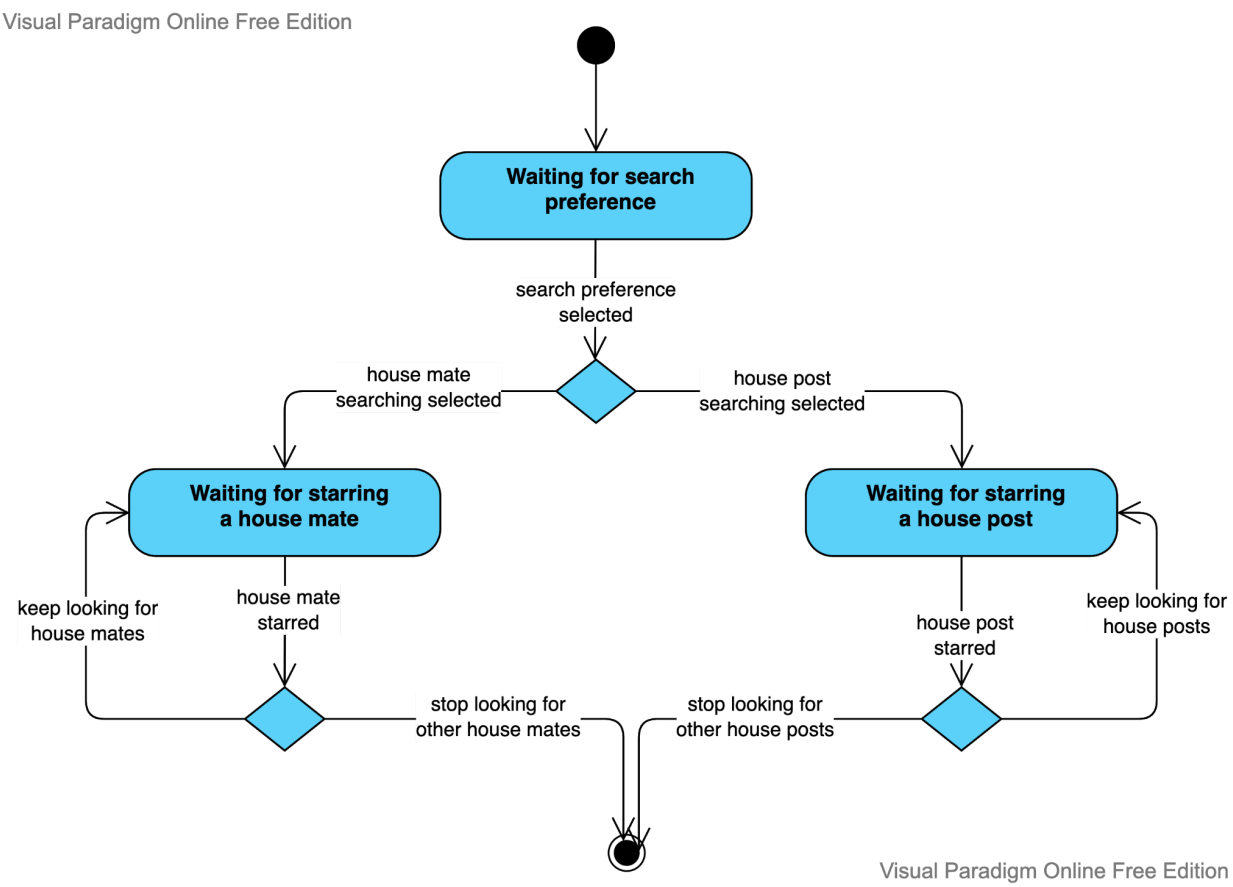


Figure 12: State diagram for "find & star post" flow

The second diagram shows the states during starring posts flow. This flow starts on the already logged in state where the user selects their preference of searching. It ends when they like one or more posts.

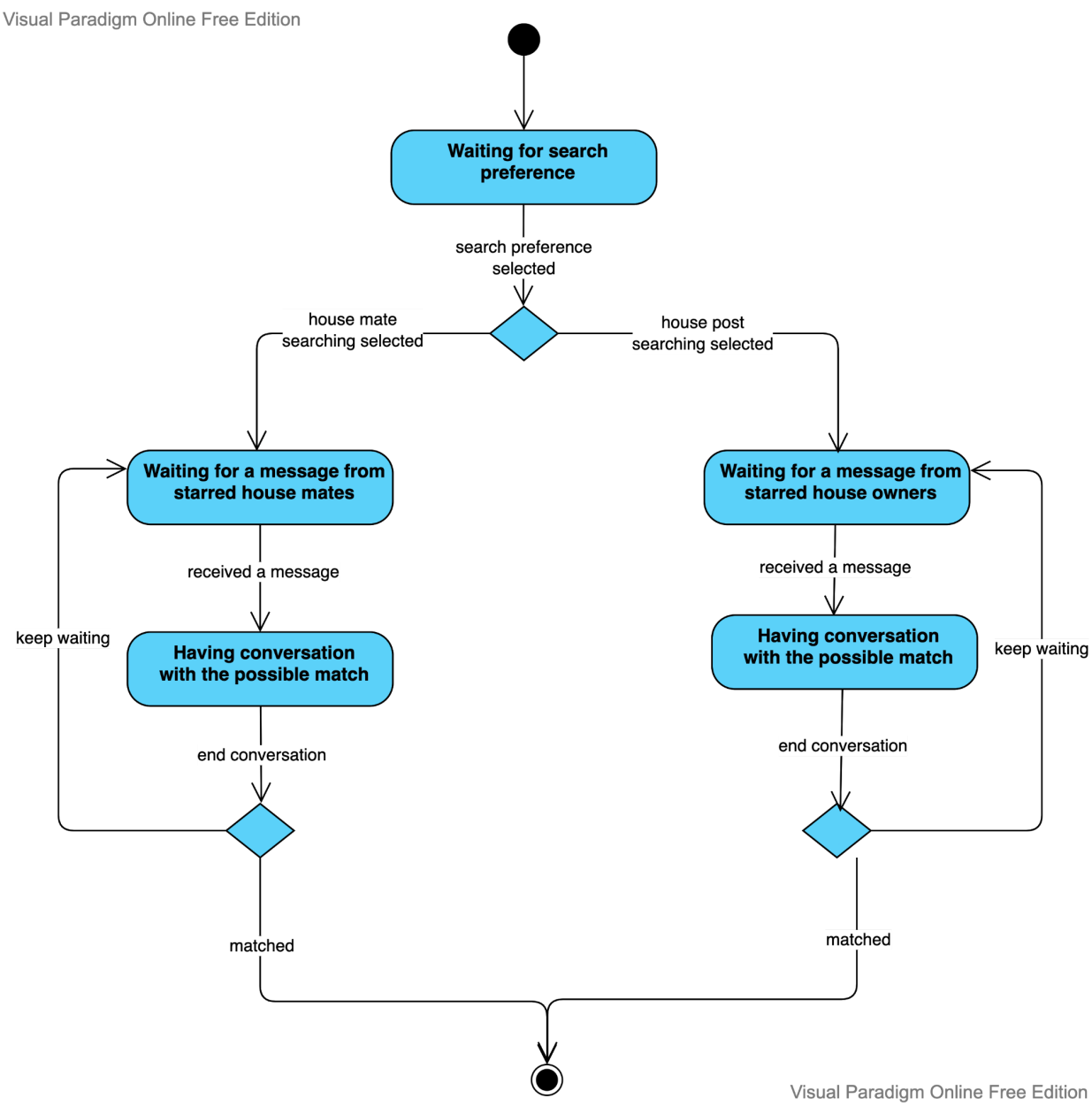
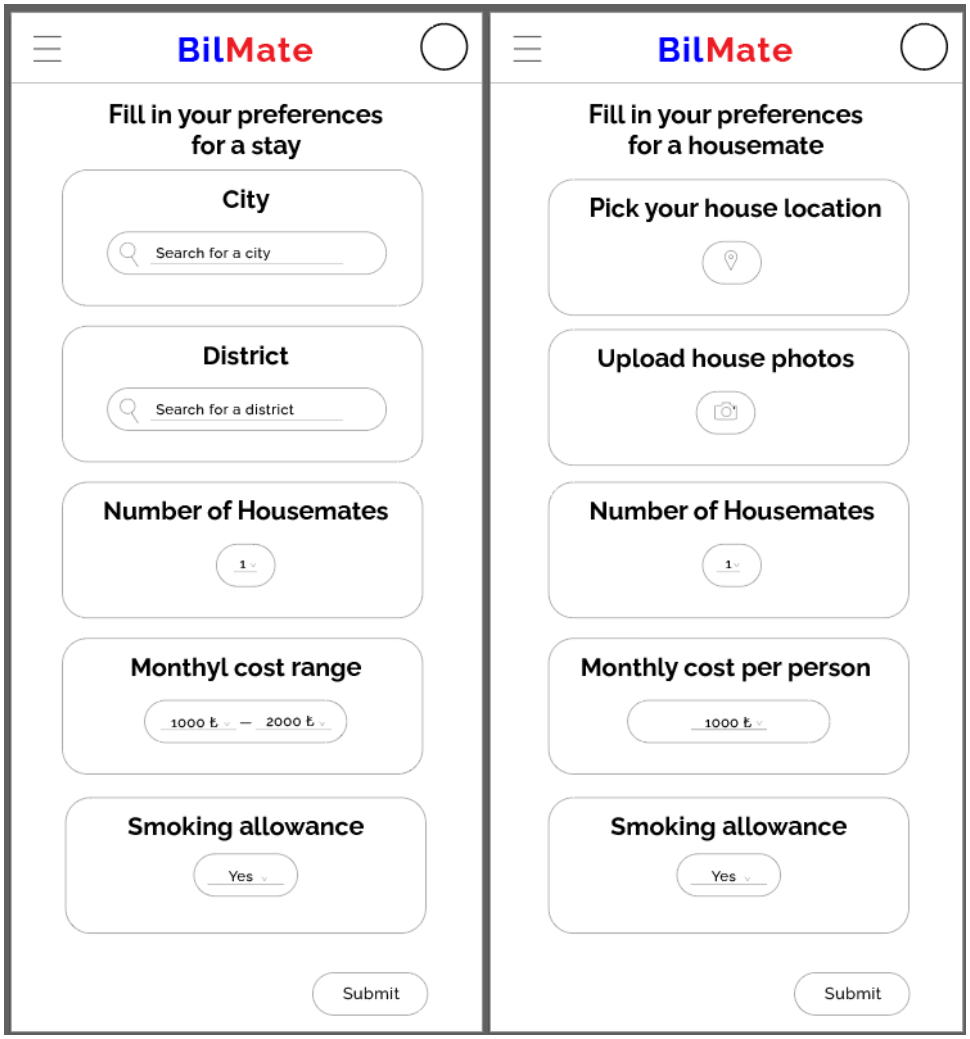


Figure 13: State diagram for "conversation with housemate/owner" flow

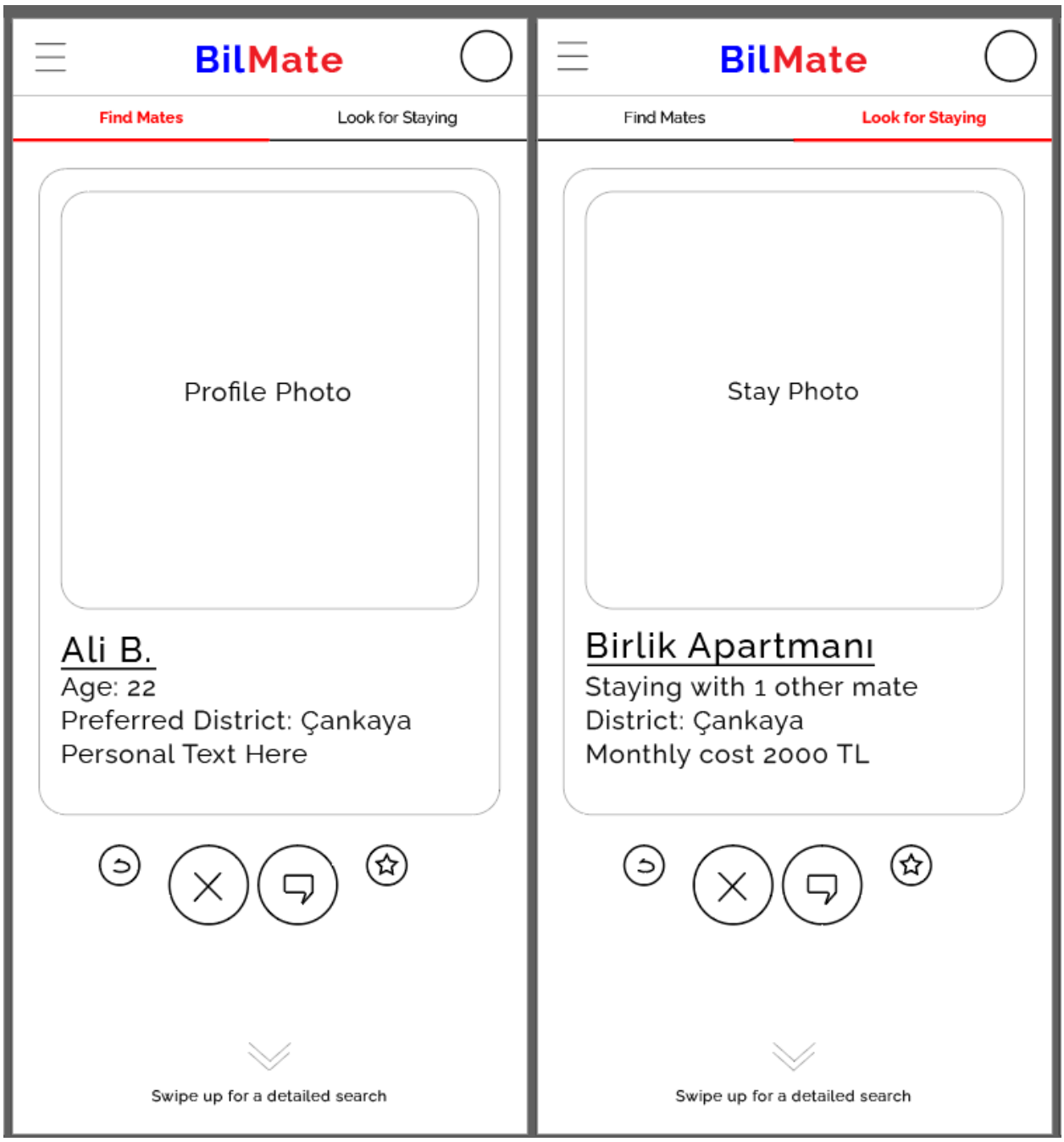
The third diagram shows the states for having a conversation with the housemate/owner via a third-party platform. It again starts in the already logged-in state, where the user selects their preference for searching. Then they wait for a message from other users. Either they agree and exit the flow, or they keep waiting for others. It is important to note that "Having a conversation with the possible match" will occur in another app's task (either email or a chatting app) that will be navigated through our app.

3.5.5 User Interface

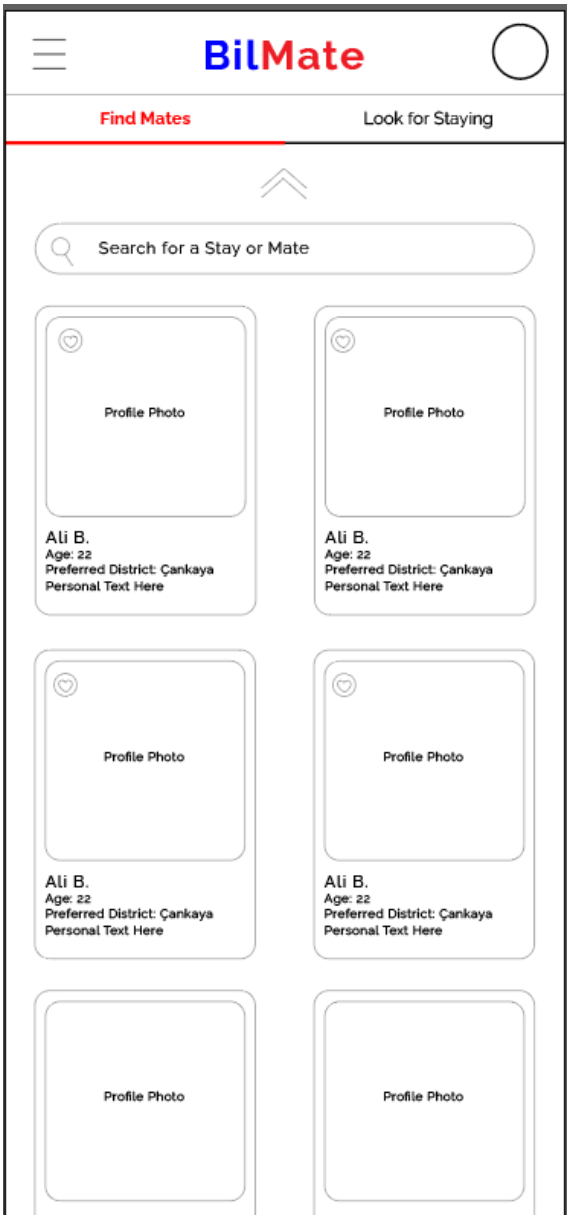
3.5.5.1 Preference Filling for House Seeking and House Sharing Upon Registration



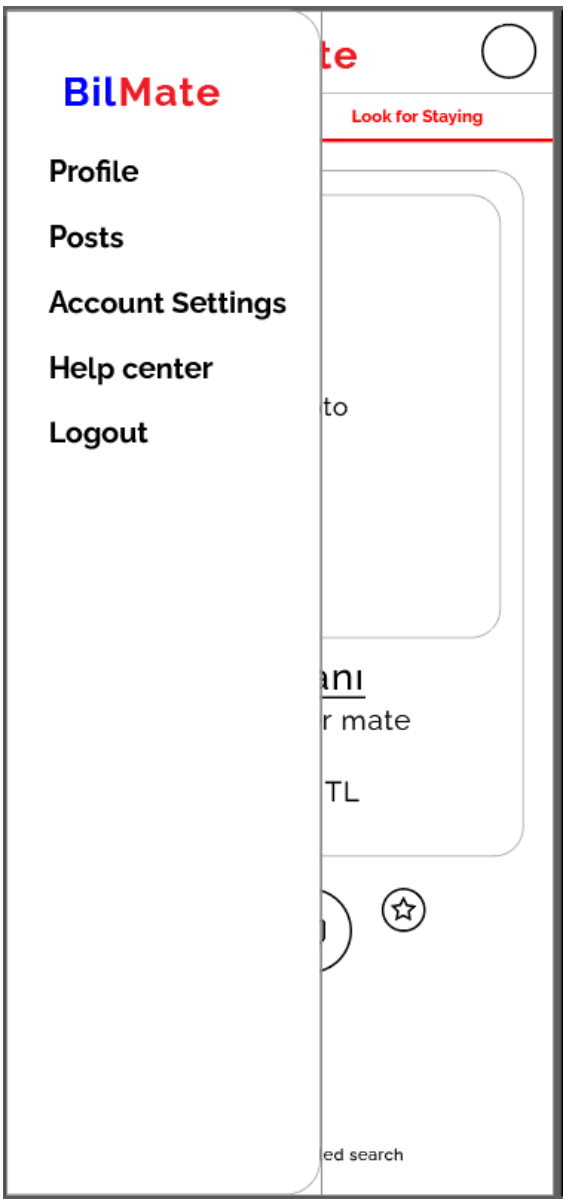
3.5.5.2 Getting Matched for House Sharers or House Seekers



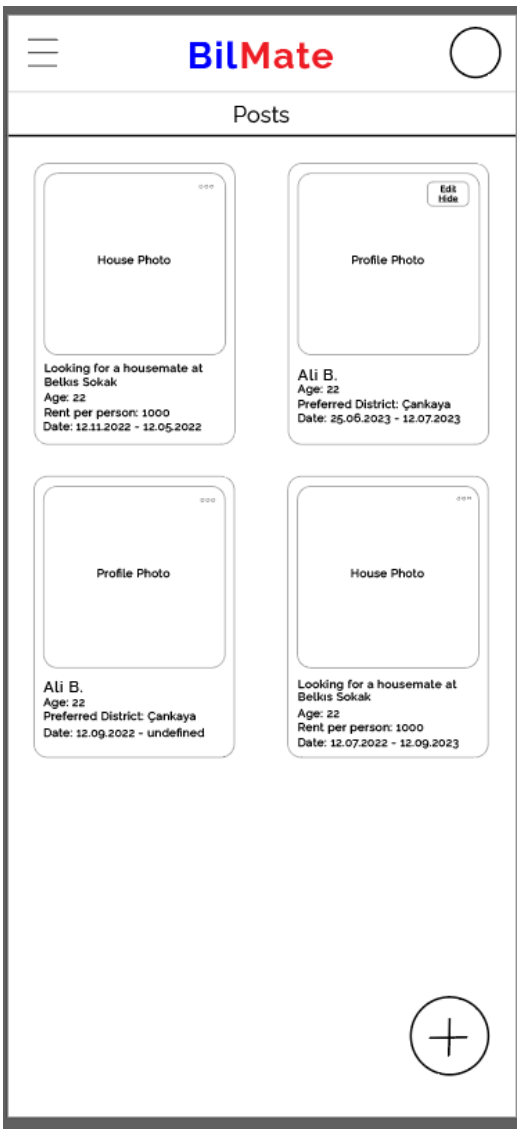
3.5.5.3 Detailed Search for House Sharers or House Seekers



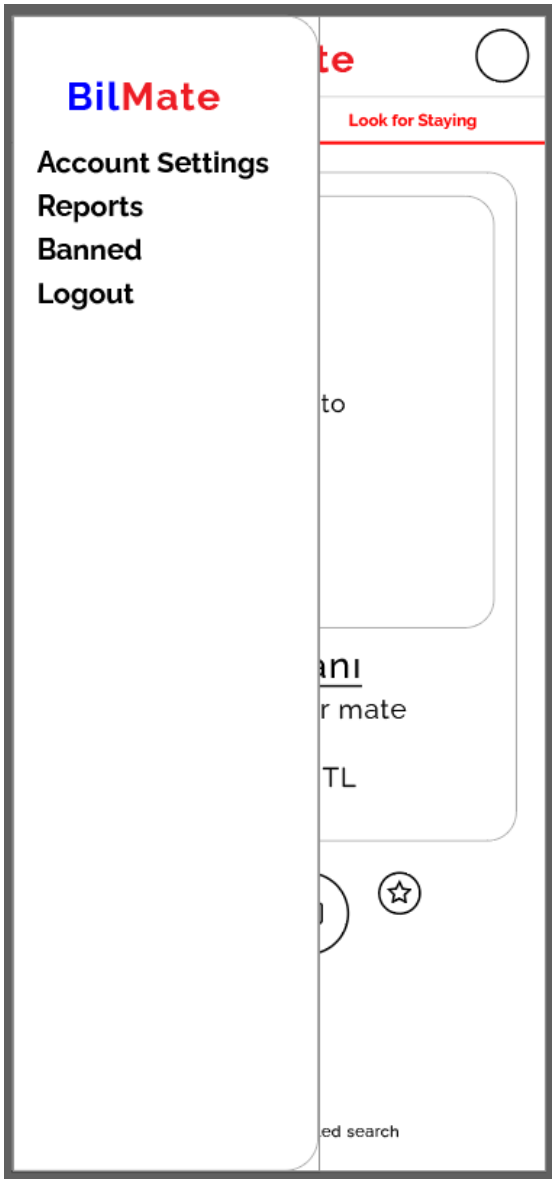
3.5.5.4 Menu


BilMate
Profile
Posts
Account Settings
Help center
Logout
te
Look for Staying
to
ani
r mate
TL
ed search

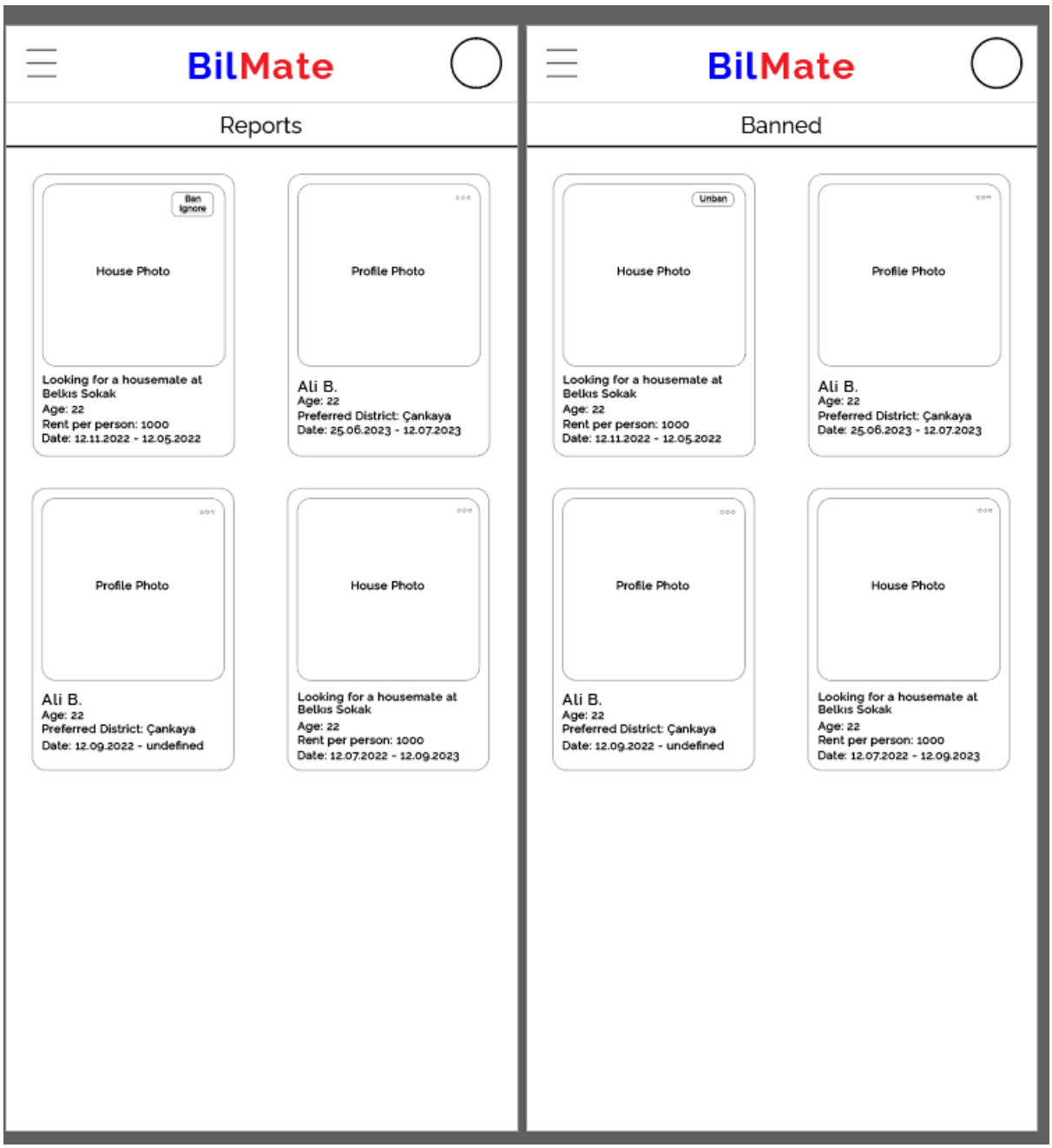
3.5.5.5 Posts for House Seeking or House Sharing



3.5.5.6 Admin Account Menu



3.5.5.7 User Reports and Banned Posts and Accounts



4.0 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

This section will cover some factors that might impact the project's development. Please find them below.

4.1.1 Public Health

Issue of public health does not affect the project. We expect users' own control in case of a health-related situation. A house-sharer and house-seeker must plan according to their current health situation.

4.1.2 Public Safety

To ensure the safety of users, a reporting feature will be added to the project. In case of encountering a malicious post, a user can report the post with a comment explaining the reasoning behind the report. This request will be taken and sent to a list of reported posts, where each post contains a counter displaying the number of times that post is reported. An admin of the application will observe each post and corresponding comments, and if feasible, the reported user and their posts will be deleted and will be banned for an indefinite duration.

Also, as part of the project's terms, no user data will be publicly shared or with an untrusted third-party app.

4.1.3 Public Welfare

Primary aim of the project is to connect students who are willing to share their house with students that are willing to stay in a student house to reduce their costs and enjoy a better accommodation environment. Besides this user life improvement, to enhance the app's usability, the application's lightness, along with a fluent interface, will be a primary design concern. Thus, comfort for users will be provided both on the application side and in real life.

4.1.4 Global

As mentioned in the previous subsection, the target audience of the project is university students who own a house and/or are willing to stay in one of those houses. Most of the universities in Turkey teach in English, but to extend the scope and ease the use for those who do not know much English, in addition to the application's primary language, which is English, Turkish might be added in later stages of development.

4.1.5 Cultural

For Turkish and many other international students, staying in a student house is an accepted habit; thus, we do not expect many issues from the cultural side. However, further preference

options could be added to the preferences list, such as users' religion, diet, etc., to enhance the output of the matching algorithm.

4.1.6 Social

The project expects a user to provide their email and phone number, which will be used to connect a house-seeker and sharer. Unless specified, both contact information of a user will be displayed publicly, but a user can hide their phone number, which is more prone to malicious acts. If they do, the only way to connect a house-sharer and seeker will be through their email addresses. This way, even though there might be a little drop in UX, users' security will be provided.

4.1.7 Environmental

Since some students will be allocated to houses rather than dormitories, this will enhance the capacity of the student dorms. This way, those who are in actual need can stay in dorms, and others can stay with some house-sharer by increasing their accommodation comfort while reducing their bills.

4.1.8 Economic

BilMate will be a free-to-use application since its launch. However, donation methods, such as "buy me a coffee" might be added to provide a way of income to developers.

	Effect Level	**Effect**
Public Health	0	We expect users' own intervention in case of a public health situation
Public Safety	8	Forbiddance of users with malicious intents, Protection of users' data
Public Welfare	7	Lightweight application, Outstanding User Interface along with User Experience
Global	2	Extend the language of the project
Cultural	1	Adding extra required preferences to the preference list
Social	8	Hiding a user's phone number, Providing a way of

		communication in case of hidden user phone number
Environmental	0	Scope of the environment is not of part of the project
Economic	1	Add donation section for a way of income

Table 1: Factors that can affect analysis and design

4.2 Risks and Alternatives

We may encounter several possible risks we may encounter throughout the implementation process and after releasing the application for this project. We should be aware of these risks and have a backup plan.

First of all, after releasing the application, new market competitors may present more advantageous offers for users. This is a likely risk we will encounter after release since after an idea has a consumer mass eager to benefit from it, many replicas of the original system will appear on the market. We should be prepared for this case from the beginning, specifying the necessary actions in such a case. Our application should offer convincing benefits for our user group to become our steady customers and free advertisers. Also, if a critical competitor appears in the market, we should take action based on the characteristics of that competitor.

Second, our application needs a minimum number of users in the first place to serve appropriately. Thus, there should be some willing users to begin housemate/roommate finding to achieve suitable matches using our application. Therefore, we should have good advertisements to reach the corresponding user group and convince them to use our application.

Third, even after a significant number of users start to use our application, there are some periods in the year when the market is inactive, especially after semesters begin and people find their roommates and houses. We should be cautious not to lose our customers, especially for the active periods of the market. There should be convincing promotions or campaigns to keep them as our customers for the future.

Fourth, there may be an abuse of the program not obeying the ethical rules, which may damage the application's image and cause us to lose some of our customers, which other users may be uncomfortable with. In such cases, we should pay attention to user feedback, review the reports very carefully, and remove the suspicious user if necessary. This action may increase our reputation in the market and for our customers.

Risk	**Likelihood**	**Backup Plan**
New competitors in the market	HIGH	Offer more benefits than competitors
Insufficient number of users	MEDIUM	Advertisement and

		promotions
Seasonal decrease in the number of users	MEDIUM	Have some promotions not to lose existing users
Abuse of the application by the users	MEDIUM	Careful review of reports

Table 2: Risks

4.3 Project Plan

This section will bring forward the Work Breakdown Structure (WBS) which outlines the general Work Packages (WP) for our project:

Milestone #	**Milestone Title**	**Members Involved**
WP_1	Project Specification Document	All Members
WP_2	Innovation Expert Meeting	All Members
WP_3	Requirements Elicitation and Analysis	All Members
WP_4	Prototype Demo	All Members
WP_5	Design Report	All Members
WP_6	Final Report	All Members
WP_7	Final Demo	All Members
WP_8	Backend Development	Ahmet Salman, Ebrar Bozkurt, Atasagun Samed Şanap
WP_9	Frontend Development	Javid Moradi, Onuralp Avcı

Table 3: List of work packages

WP_1: Project Specification Document	
Start date: *15.09.2022* **End date:** *17.10.2022*	
Members involved:	*All Members*
Objectives: *Choosing a topic which is innovative and tackles complex engineering challenges, and deciding on the feature set we would like to bring to our clients*	
Tasks: ***Task 1.1 Identifying constraints*:** *Analyze the constraints for our project* ***Task 1.2 Identifying requirements* :** *Determine the requirements with respect to our client's expectations* ***Task 1.3 Identifying ethical and legal issues* :** *Research possible ethical and legal issues surrounding the storage of our users' date*	

WP_2: Innovation Expert Meeting	
Start date: *15.09.2022* **End date:** *17.10.2022*	
Members involved:	*All Members*
Objectives: *Meeting with our innovation expert in order to get a better view of the requirements*	
Tasks ***Task 2.1 Discussing Improvements*:** *Enrich our feature set and ensure optimal user experience* ***Task 2.2 Monetization Strategy* :** *Analyzing the methods which we can cover server costs and continue serving our customers* ***Task 2.3 Researching Possible Pitfalls* :** *research possible obstacles that could prevent our application from smooth operation*	

WP_3: Requirements Elicitation and Analysis	
Start date: *17.10.2022* **End date:** *07.11.2022*	
Members involved:	*All Members*
Objectives: *Designing our system using UML diagrams and stating ethical, professional, and engineering goals*	
Tasks: ***Task 3.1 Refining Functional Requirements*:** *Finazlizing what each user can do including the admins and reporting system*	

Task 3.2 Researching Past Systems: *Look into the disadvantages of past systems and their shortcomings in order to offer a better product*

Task 3.3 Finalizing Feature Set**:** *Finalizing the mechanisms that users can use to interact with our application*

Task 3.4 Matching Algorithm Initial Design**:** *Decide on the method the matching algorithm should work*

WP_4: Prototype Demo	
Start date: *07.11.2022* **End date:** *17.12.2022*	
Members involved:	*All Members*
Objectives: *Presenting our progress and future plans to the jury and our classmates*	
Tasks: ***Task 4.1 Identifying the Problem*****:** *Presenting the problem statement, why do we need BilMate* ***Task 4.2 Presenting the Solution*** **:** *Stating how BilMate can help solve the aforementioned problem* ***Task 4.3 Stating what has yet to be completed*** **:** *Mention the shortcomings of the current implementation and how we plan to fix them*	

WP_5: Design Report	
Start date: *15.01.2023* **End date:** *20.02.2023*	
Members involved:	*All Members*
Objectives: *Reconsider the changes in requirements and design and reflect on them*	
Tasks: ***Task 5.1 Refining Design*****:** *correct the design to account for changes that occurred in the development process*	

WP_6: Final Report	
Start date: *20.02.2023* **End date:** *15.05.2023*	
Members involved:	*All Members*
Objectives: *Finalize both the design and the application*	
Tasks: ***Task 6.1 Ensure Consistency*****:** *Compare the final design documents with the program we developed to ensure they are consistent*	

Task 6.2 Ensure Correctness **:** *After confirming that the program design adheres to the design document, ensure it is bug-free*

WP_7: Final Demo	
Start date: *20.02.2023* **End date:** *15.05.2023*	
Members involved:	*All Members*
Objectives: *Present the finalized product to the jury and our classmates*	
Tasks: ***Task 7.1 Preparing the presentation*****:** *Prepare the presentation with the proper pitch and time constraints* ***Task 7.2 Identifying requirements*** **:** *Determine the requirements with respect to our client's expectations* ***Task 7.3 Identifying ethical and legal issues*** **:** *Research possible ethical and legal issues surrounding the storage of our users' date*	

WP_8: Backend Development	
Start date: *15.09.2022* **End date:** *15.05.2023*	
Members involved:	*Ahmet Salman,* *Ebrar Bozkurt,* *Atasagun Samed Şanap*
Objectives: *Build the server-side of our program*	
Tasks: ***Task 8.1 Basic Functionalities*****:** *Develop basic functionalities such as user registration and post adding* ***Task 8.2 Security*** **:** *Ensure that the data stored in our Database is secure and hardened against attacks* ***Task 8.3 Matching Algorithm*** **:** *Develop the matching algorithm which will be used to accurately match house-sharers and house-seekers* ***Task 8.4 Database Schema Design*** **:** *Design the Database that will hold all the data of our users*	

WP_9: Frontend Development	
Start date: *15.09.2022* **End date:** *15.05.2023*	
Members involved:	*Javid Moradi,* *Onuralp Avcı*

Objectives: *Develop a responsive and appealing UI for the program*
Tasks: ***Task 9.1 Ensuring Responsiveness*** **:** *Develop the UI using strategies that ensure responsiveness and fast response* ***Task 9.2 Visual Appeal*** **:** *Utilize UI libraries to create an appealing and user-friendly interface for the users*

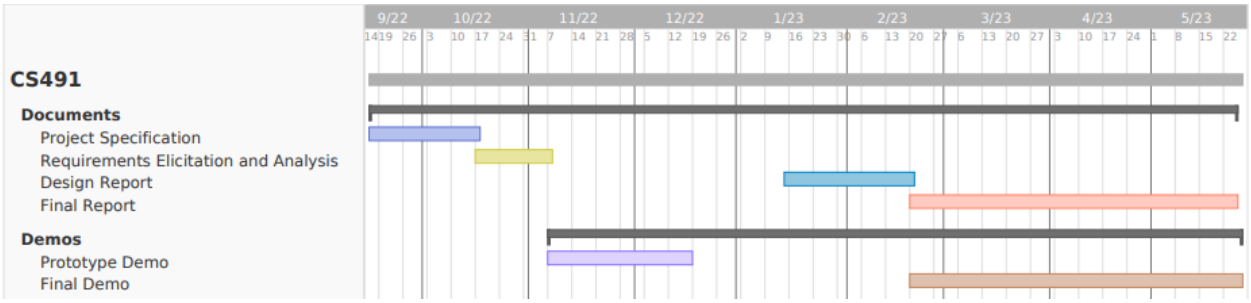


Figure 14: Gantt Chart for our project

4.4 Ensuring Proper Teamwork

In order to ensure fair work division and proper teamwork among us, we have decided to use Trello as a general project management tool. Moreover, in Trello, we have created “to-do”, “doing”, and “done” columns which give a transparent view of who is doing what and how the work is divided.

In addition to fair work allocation, it is also important to ensure that the work is up to a high standard, and thus it needs to be reviewed. As such, from the beginning, each person was assigned 2 team members with who he/she should review their work.

Another method to ensure smooth sailing in the teamwork department is the agile-like weekly meetings in which we discuss the current objectives, future milestones, and possible obstacles.

4.5 Ethics and Professional Responsibilities

- We will follow the ACM Code of Ethics and Professional Conduct throughout the project [1].
- Since this application is designed to be used by students, login of non-students should not be allowed by verification of being a student. This verification will be done by enabling signup with only university emails.
- The users' data should not be shared with third-party companies without the user's permission.
- In the case of abuse and misuse of the program, such as acts of violence and posting irrelevant postings, the involved users are banned from the system and will not be allowed to use the application again.

- The data of the users should be considered sensitive data and be kept secure using appropriate encryption algorithms, and the application should use a “data protection by design and default” strategy.

4.6 Planning for New Knowledge and Learning Strategies

A learning plan for the technologies that will be used for the project is initially familiarize with the language or the framework by first conducting a literature review on the related technology and trying to produce a simple and explanatory mini-project; this is done by watching a short crash course of the technology. After familiarizing with the overall workflow, to expand the knowledge, further analysis and research will be done on a specific topic of the technology. So, in general, a literature review, mainly on the world wide web, will be the main course of action as a learning strategy.

In addition, for the server side, since the load of the concurrent could be high, load balancing is a must. To achieve this, a literature review will be conducted, and different approaches for load balancing will be analyzed. For the client side, to ensure the code is well-written, the Data-Domain-UI layers design pattern will be used. Also, to enhance writability and readability, different design patterns will be utilized for both the back and front-end sections of the project.

6.0 References

[1] ACM Code of Ethics and Professional Conduct. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 05 November 2022]