

Software Requirements Specification

Of

Roommate

**Shrivathsa Keshava Murthy
Chaithanya Padubidri Nagaraja Rao
Athreya Karkada Nakshathri**

REQUIREMENT SPECIFICATION

University of Colorado, Boulder

Table of Contents

Table of Contents

1. Project Summary

2. System Requirements

Business requirements User requirements Functional requirements
Non-Functional requirements

3. Use case View

Use case Diagrams Use Case Descriptions

4. Logical View

Class Diagrams Activity diagrams Sequence diagrams

5. UI Mockups

6. Data Storage

1. Project Summary

We all have problems finding room-mates as there is no common platform to search. This Java based web application makes finding compatible roommates easier. Users can sign up through a unique email address or using Facebook login that queries their respective profiles. The application has a database which stores registered users' information. When a user wants to find a room-mate, our algorithm finds the closest match based on profile rank, history and interest. Once a room-mate is found, there is an option to select and rate

2. System Requirements

Business Requirements			
ID	Requirements	Topic Area	Priority
BR-001	Users must enter a valid email id in order to register themselves. The user will receive a verification email and upon verification, will be added to the database	Authentication	Critical
BR-002	Users must complete all the required fields while building the profile	Profile Creation	Critical
BR-003	There cannot be two users with the same email-id	Validation	Critical
BR-004	Users must accept the privacy terms and conditions in order to use the application	Disclaimer	Critical

User Requirements			
ID	Requirements	Topic Area	Priority
UR-001	A user must be able to search for other roommates based on name or email-id and send roommate requests	Authentication	Critical
UR-002	Only upon accepting the recommendation/requests, a user must be able to chat with the accepted roommate recommendation	Profile Creation	Critical
UR-003	Users must be able to decline/accept the	Approval	Medium

	recommendation		
UR-004	User must be able to view a history of recommendations and searches	Archive	Medium
UR-005	Users must be categorized as one with the place or looking for a place	Category	Critical
UR-006	A user must be able to edit his profile	Modify	Critical
UR-007	When a roommate leaves the house, users must be able to rate each other	Feedback	Medium
UR-008	A user must not be able to see his own rating, but others' rating must be visible	Privacy	Medium
UR-009	User must be able to get notifications about the matched roommates and the criteria based on matching	Notification	Critical
UR-010	User may be able to write a testimonial about the application	App Feedback	Medium
UR-011	New user/Existing user must be able to see the main menu after successful login	Post-login	High
UR-012	Existing user must be able to see his saved profile	Post-login	High
UR-013	User must be able to successfully log out after hitting the log-out button	App Exit	Critical
UR-014	User must be able to filter from the suggested roommates	Filter	Critical
UR-015	User must be able to recover the password	Recovery	Critical
UR-016	User must be able to sign up using Facebook account	Authentication	Critical

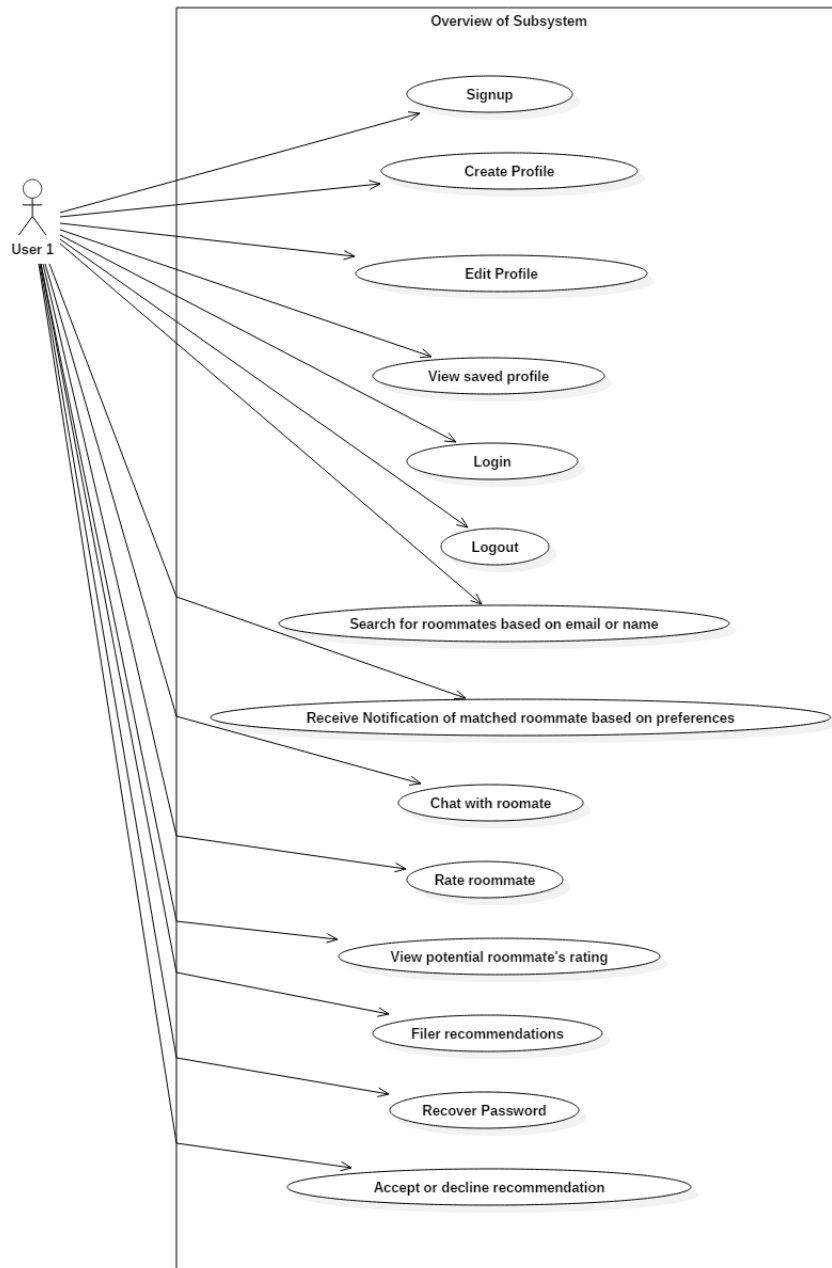
Functional Requirements			
ID	Requirements	Topic Area	Priority
FR-001	The system must be able to match profiles based on location first	Algorithm	Critical
FR-002	The system must maintain 2 IDs for each user, one individual and one upon matching roommates (and	User identification	Critical

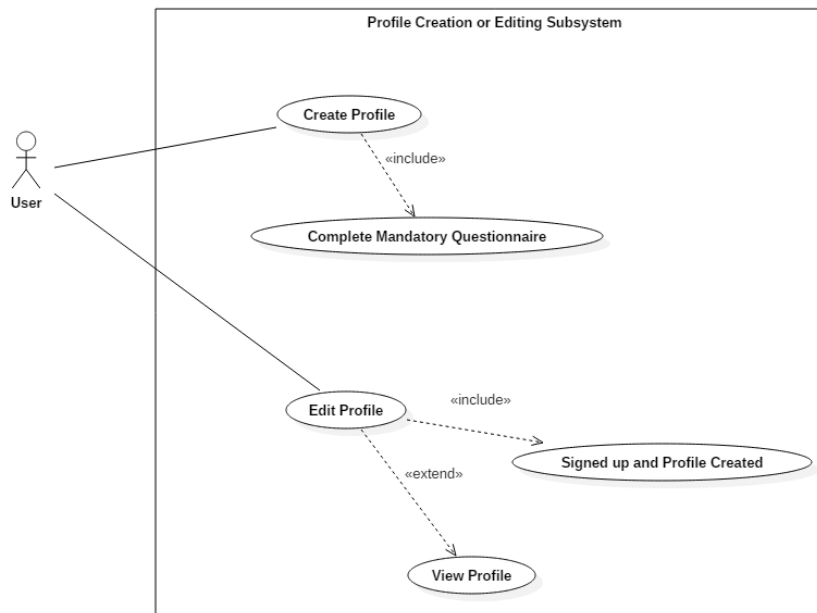
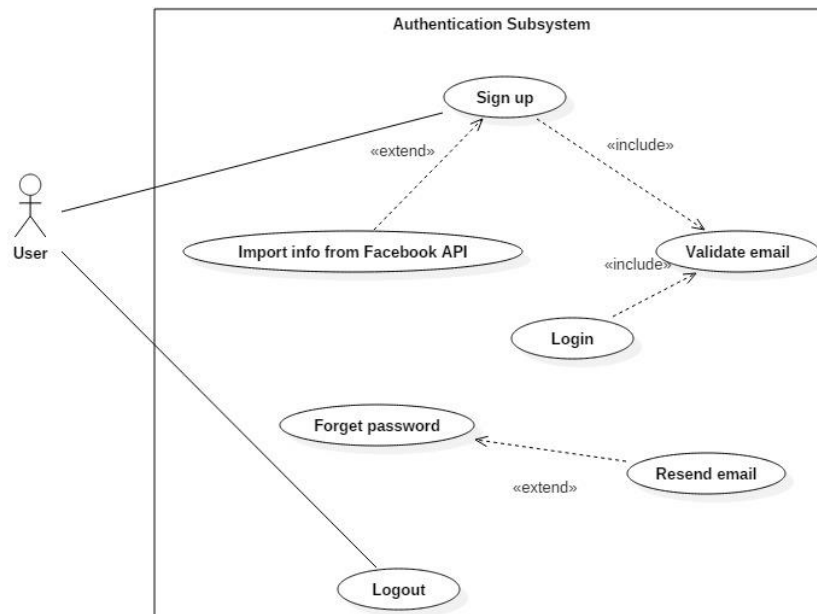
	when they're living together, a group ID)		
FR-003	The above 2 IDs must be maintained in 2 separate tables in databases.	Tracking IDs	Medium
FR-004	The system must maintain each user in either an unmatched database or matched database, while in the unmatched database, no group ID must be assigned	User-database association	Medium
FR-005	When a person exits a group, the group ID should no longer be associated and that person must be put back in the unmatched database, and notifications must be sent to each user in the group to rate the exiting person from the group and also ask the other users with the group ID, if the system should run the matching process again to look for another new roommate.	Roommate-exiting process	Critical
FR-006	System must match a user without a place with users with a place, in and around the same location	Matching algorithm	Critical
FR-007	System must be able to pull data from the appropriate databases	Data exchange	Medium
FR-008	System should display user testimonials at the home page before logging/signing up	Feedback	Medium
FR-009	System should log out a user upon 30 minutes of inactivity	User activity tracking	Medium

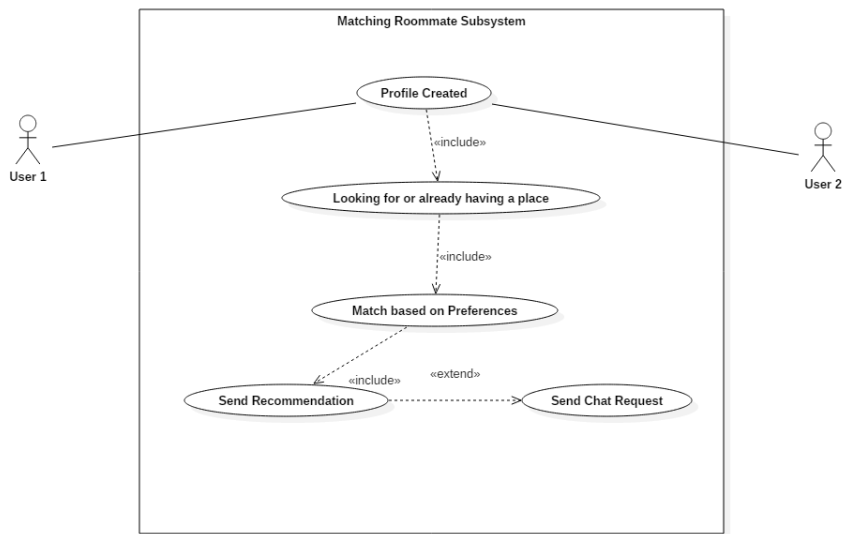
Non-functional Requirements			
ID	Requirements	Topic Area	Priority
NBR-001	The application should be accessible at all times without almost zero downtime	Availability	Critical
NBR-002	The functionality of the system should not be affected by any component upgrade	Sustainability	Critical

3. Use Case View

Use Case Diagrams







Use Case Documents:

Use Case ID	UR-001
Use Case Name	Roomie Search
Description	A user must be able to search for other roommates based on name or email-id and send roommate requests

Actors	User (Roommate Seeker)		
Pre-conditions	Seeker should have logged in with valid credentials. The other person being searched should already have an entry in the database		
Post-Conditions	Roomie seeker successfully finds a match based on username/mail-id		
Frequency of Use	Frequently throughout the day by the user		
Flow of events		Actor Action	System Response
	1	Enter username and password	Authenticate the user
	2	Search for a user match based on username/mail-id	Return a successful response if a valid entry for the user is found in the database
Variations			

Exceptions	<ol style="list-style-type: none"> 1. Incorrect login information – Fails to enter main menu 2. No database entry – System fails to find a match from the database
Notes and Issues	
Developer Notes:	

Use Case ID	UR-002
Use Case Name	Match validation
Description	Only upon accepting the recommendation/requests, a user must be able to chat with the accepted roommate recommendation

Actors	User		
Pre-conditions	<ul style="list-style-type: none"> • User has signed in. • Made a query based on preference. • Got a match based on his query • Accepted the match 		
Post-Conditions	User can now chat with the potential roomie.		
Frequency of Use	Frequently throughout the day by the users		
Flow of events		Actor Action	System Response
	1	Enter username password	Authentication
	2	Search for a match based on preference	Database query

	3	Click on accept match	Display the chat window
	4	Chat with the potential roomie	
Variations	User finds a match based on his preference. But refuses to confirm the match. Chat window will not pop-up		
Exceptions			
Notes and Issues			
Developer Notes:			

Use Case ID	UR-003
Use Case Name	Recommendation Approval
Description	Users must be able to decline/accept the recommendation

Actors	Roommate Seeker
Pre-conditions	Roommate seeker must be logged in to the app Must have updated the profile based on preferences Must have got matches
Post-Conditions	Of the available matches seeker can decline/accept matches
Frequency of Use	High

Flow of events		Actor Action	System Response
	1	Seeker must have logged in	
	2	Once the match happens seeker can decline the match	System generates list of possible matches based on seeker's preferences and provides the suggestion If user declines, then those matched entries will be removed
Variations			
Notes and Issues			
Developer Notes:			

Use Case ID	UR-004
Use Case Name	Search History
Description	User must be able to view a history of recommendations and searches

Actors	User (Roommate Seeker)
Pre-conditions	User is logged in User is on the home page User has searched for a potential roommate based on preference

	User has found more than one matches based on preferences		
Post-Conditions	User saves the matches		
Frequency of Use	Medium		
Flow of events		Actor Action	System Response
	1	User updates his profile based on preferences	Records user's updates
	2	User clicks on find match button	Finds matching entries based on user criteria
	3	User saves all potential matches	Updates filtered matches for that particular user
Variations	From the list of saved matches, user can delete certain matches and narrow down his selection		
Exceptions	User has entered incorrect login credentials. System fails to identify the user.		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-005
Use Case Name	Roomie categorization

Description	Users must be categorized as one with the place or looking for a place
-------------	--

Actors	Roommate seeker		
Pre-conditions	User must be logged in to the application User must click on edit profile page User updates information about whether he/she has a place to stay		
Post-Conditions	User finds matches based on this criteria		
Frequency of Use	Frequently throughout the day		
Flow of events		Actor Action	System Response
	1	User logs in to the application	Authentication
	2	User clicks on edit profile	
	3	User updates his place information	System finds matches based on this preference
	4	User finds roommates based on this preference	
Variations			
Exceptions	Login failure. Database failure		
Notes and Issues			
Developer Notes:			

Use Case ID	UR - 006
Use Case Name	Update profile
Description	A user must be able to edit his profile

Actors	Roommate Seeker		
Pre-conditions	User must have successfully signed in User must enter his preferences		
Post-Conditions	Users profile preferences are saved		
Frequency of Use	Frequently throughout the day		
Flow of events		Actor Action	System Response
	1	User logs in to the application	Authentication
	2	User clicks on update profile	Database entries for that user-id are saved
Variations	User can delete his profile and create a fresh one		
Exceptions	Failure to log-in. Information is not saved into the database since user fails to enter all the valid fields.		
Notes and Issues			

Developer Notes:	
------------------	--

Use Case ID	UR-007
Use Case Name	Roomie Rating
Description	When a roommate leaves the house, users must be able to rate each other

Actors	Roommate Seeker		
Pre-conditions	Roommate logs in Roommate updates profile Roommate finds a match Roommate finalizes potential roommates Roommate changes his location		
Post-Conditions	Other roommates registered should now be able to rate the roommate		
Frequency of Use	Moderate		
Flow of events		Actor Action	System Response
	1	User logs in	
	2	User updates his profile and finds a potential roomie match	
	3	User leaves the place/changes his location in the app	Update user's location info. Display an option for other roommates to rate the departing roommate

Variations	
Exceptions	
Notes and Issues	
Developer Notes:	

Use Case ID	UR-008
Use Case Name	Privacy
Description	A Roommate seeker must not be able to see his own rating, but others' rating must be visible

Actors	Roommate Seeker		
Pre-conditions	Roommate seeker must be logged in Roommate seeker must click on 'Search Roommates' option		
Post-Conditions	Roommate seeker must be able to see other potential roommates rating		
Frequency of Use	High		
Flow of events		Actor Action	System Response
	1	Roommate seeker updates his preferences and clicks on the search button	System records seeker's preferences and finds matches

	2	Roommate seeker is able to see other potential roommate's rating and make a decision	System fetches potential roomies information from the database along with ratings. Hides the seeker's rating information
Variations	1. Roommate seeker should be able to sort his potential matched roomies based on ratings		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-009
Use Case Name	Notification
Description	User must be able to get notifications about the matched roommates and the criteria based on matching

Actors	Roommate Seeker		
Pre-conditions	Roommate Seeker must have successfully logged in Must have searched for potential matches based on preferences. Should have clicked on send notification updates		
Post-Conditions	Notifications via Email are sent		
Frequency of Use	Moderate		
Flow of events		Actor Action	System Response

			System records roommate seeker preferences, finds a match stores it as a key value pair for the roommate seeker
		Roommate seeker clicks on send notifications	Using seeker's mail-id, system sends mails for every potential match
Variations	Seeker can click on disable notifications to stop sending any. Adjust frequency of mails.		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-010
Use Case Name	Testimonials
Description	User may be able to write a testimonial about the application

Actors	Roommate Seeker		
Pre-conditions	Roommate seeker logs in to the application. Clicks on write testimonials and submit		
Post-Conditions	Testimonials written by all seekers are displayed on the main page of the application before the login screen		
Frequency of Use	Moderate		
Flow of events		Actor Action	System Response

		Seeker clicks on write testimonials	System records the text based testimonials from the seekers mapped to their user id's
			Saves it in the database, displays it in the homepage before the login option
Variations	User can delete his testimonial, rate testimonial as helpful if needed		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-011
Use Case Name	Main Menu
Description	New user/Existing user must be able to see the main menu after successful login

Actors	Roommate Seeker
Pre-conditions	Seeker should have entered valid login credentials Returning users must click on log-in page New users should click on sign-up page
Post-Conditions	Upon successful login returning/new users should see the application home page
Frequency of Use	High

Flow of events		Actor Action	System Response
	1	Enter valid username/password if returning user and signup details for new user	System checks for a valid entry in the database for authentication and for new user creates a new entry
	2	Enter the Home page	Successful validation should prompt the user to go to the home page
Variations	Authentication failure. Ask for re-login		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-012
Use Case Name	Saved Profile
Description	Existing user must be able to see his saved profile

Actors	Roommate Seeker
Pre-conditions	Seekers should enter valid credentials for returning users Click on profile page
Post-Conditions	Should be able to see saved preferences

Frequency of Use	High		
Flow of events		Actor Action	System Response
	1	Enter valid username/password	Authenticate
	2	Returning users	
	3	Click on preferences	Populate preferences
Variations	Invalid credentials, fails to load preferences		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-013
Use Case Name	Logout
Description	User must be able to successfully log out after hitting the log-out button

Actors	Roommate seeker(user), admin		
Pre-conditions	Actor has already logged in and now is in the logout page		
Post-Conditions	Successfully logs out and clears the session		
Frequency of Use	High		
Flow of events		Actor Action	System Response

	1	User logs in	Authenticate
	2	Hits logout	Logout session
Variations	Fails to logout, still in the main menu		
Notes and Issues			
Developer Notes:			

Use Case ID	UR-014
Use Case Name	Filter suggestions
Description	User must be able to filter from the suggested roommates

Actors	User(roommate seeker)		
Pre-conditions	User has already logged in, created profile, made a search, has got some responses		
Post-Conditions	Flags some of the suggested responses		
Frequency of Use	Medium		
Flow of events		Actor Action	System Response
		Log in	Authenticate
		Create profile	Write to DB

		Make a new query	Has provided responses to user
		Flags his options	Saves it successfully for future
Variations	1. Failed login 2. If there is no match at all, this case does not come into picture		
Notes and Issues			
Developer Notes:			