Test Cases

RoomieLah: A Roommate Matching Mobile Application

Delivered By:
StrawHats

Arora Srishti
Pandey Pratyush Kumar
Surawar Sanath
Iyer Rajagopal Mahadevan
Acharya Atul
Tayal Aks
Agarwal Gopal

Submitted To -
Shi Hanyu
School of Computer Science and Engineering
Nanyang Technological University

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Arora Srishti	06/04/2021	Tayal Aks	07/04/2021	Initial Test Cases

Table of Contents

Unit Testing

The primary purpose of unit Testing is to test the individual pieces and components of RoomieLah. When the procedures are tested, they are deemed to be fit to use in the application.

Test Case 1

Test Case #	UT1	**Test Name**	Bottom Navigation
System	RoomieLah	**Sub System**	Home Page
Designed By	QA Team	**Design Date**	18 March 2022
Executed By	QA Team	**Execution Date**	28 March 2022
Description	There are three buttons in the bottom navigation bar: Home (View recommendations), View/Edit Profile and Chats		

Pre - Conditions	The mobile device must be connected to a stable internet connection The user must be logged in to RoomieLah using their email account

Step #	**Action**	**Expected System Response**
1	Tap on the ‘Home’ icon of the bottom navigation bar.	The Home page is displayed where different user profiles are appearing. The user is then able to swap left and right to choose a potential match.
2	Tap on the ‘Edit’ icon of the bottom navigation bar.	The user’s current profile information is displayed with an option to edit the details and personal and roommate preferences.
3	Tap on the ‘Chat’ icon of the bottom navigation bar	The chat page is opened which displays a list of names of your matches along with your last message with them (if any).

Post Conditions	1. The respective pages - recommended profiles, view/edit profile or chats are displayed when the respective buttons are clicked.

Test Case 2

Test Case #	UT2	**Test Name**	Recommend Profiles
System	RoomieLah	**Sub System**	Home Page
Designed By	QA Team	**Design Date**	18 March 2022
Executed By	QA Team	**Execution Date**	28 March 2022
Description	The profile cards of recommended roommates are displayed on the home page. The user must be able to swipe left and right on the card to see all the recommended options.		

Pre - Conditions	1. The mobile device must be connected to a stable internet connection 2. The user must be logged in to RoomieLah using their email account 3. The user is currently on the View Recommendations (Home) page.

Step #	**Action**	**Expected System Response**
1	Swipe left on the profile card using the single finger gesture.	Next profile is displayed and the current profile is never shown again because the system records that the user is not interested in the given profile.
2	Swipe right on the profile card using the single finger gesture.	The given profile is added to the list of liked profiles maintained in the database. If the like is mutual, i.e. the given profile's user has already liked the user then a pop up is

		displayed with the words “ It’s a match”.

Post Conditions	If it is a match, the given profile is added to the list of matched profiles and is visible in the chat section.

Test Case 3

Test Case #	UT3	**Test Name**	Edit / View Your Profile
System	RoomieLah	**Sub System**	Edit / View Profile Page
Designed By	QA Team	**Design Date**	18 March 2022
Executed By	QA Team	**Execution Date**	28 March 2022
Description	The users are allowed to edit their personal details, preferences and roommate preferences on the edit/view profile page.		

Pre - Conditions	1. The mobile device must be connected to a stable internet connection 2. The user must be logged in to RoomieLah using their email account 3. The user is currently on the View / Edit Profile page.

Step #	**Action**	**Expected System Response**
1	The user clicks on the view profile button of the bottom navbar.	The user’s current profile and their preferences are displayed.
2	The user clicks on the edit button on the top right corner of the page.	The existing details become modifiable. The user is able to overwrite existing information.
3	The user clicks the “save” button after making changes.	The database is updated with new information. “Your changes have been saved

		successfully.” is displayed to the user.

Post Conditions	If an edit has been made and saved, the database is updated, and the profiles recommended to the user are refreshed according to the new preferences.

Test Case 4

Test Case #	UT4	**Test Name**	Chat
System	RoomieLah	**Sub System**	Chat Page
Designed By	QA Team	**Design Date**	18 March 2022
Executed By	QA Team	**Execution Date**	28 March 2022
Description	The chat feature allows the user to exchange real time text messages with another user who has expressed interest in being a roommate.		

Pre - Conditions	1. The mobile device must be connected to a stable internet connection 2. The user must be logged in to RoomieLah using their email account 3. The user is currently on the Chat page. 4. The user is matched with at least one other user.

Step #	**Action**	**Expected System Response**
1	From the list of users displayed on the chat page, click on one name.	The chatting history is displayed which includes messages exchanged previously and the date & time of sending/receiving.
2	Click on the white textbox at the bottom of the screen.	A virtual keyboard pops up.
3	Type out a message and hit “send”.	The message is delivered to the other user. You can see the sent message in your conversation / chat history.

Post Conditions	The sent message is uploaded from the user's device to the cloud database and downloaded to the second user's device and displayed.

Integration testing

An incremental top-down approach has been used to conduct integration tests on RoomieLah. This ensures that usability is not compromised for edge cases and that the client and server sides have been integrated properly without the occurrence of any errors or server crashes.

Test Case 1

Test Case #	IT1	Test Name	Login Integration
System	RoomieLah	Sub-System	Authentication
Designed by	QA Team	Design Date	7 April 2021
Executed by	QA Team	Execution Date	8 April 2021
Description	The login page of the system must be integrated with Firebase Auth API. Authenticated users should be allowed to access the application.		

Pre-conditions	1. At least one registered user with valid login credentials should exist on Firebase 2. Backend application server has been deployed and is listening to client requests.

Step #	Action	Expected System Response
1	Incorrect User credentials are entered i.e., email and password	System displays a pop up saying: Invalid Credentials. Please try again.
2	Correct credentials for a registered user are entered	System is redirected to the home page on the Recommendations tab.

Post-conditions	The user is on the view recommendations page.

Test Case 2

Test Case #	IT2	Test Name	Sign Up Integration
System	RoomieLah	Sub-System	Authentication
Designed by	QA Team	Design Date	7 April 2021
Executed by	QA Team	Execution Date	8 April 2021
Description	A user must be able to register on the application using Firebase Auth and Cloud services.		

Pre-conditions	1. Backend application server has been deployed and is listening to client requests.

Step #	Action	Expected System Response
1	User selects Register link and enters the credentials i.e. username, email and password.	System is redirected to the surveys page where questions about user preferences and profile are displayed. The user credentials should be reflected in the ‘Authentication’ tab of firebase dashboard.
2	The user answers the survey questions about their preferences and personal details.	The user is registered and taken to the next screen. A document with the user’s unique id should also created and added to the ‘users’ collection in the database to store user related data.

Post-conditions	A new user account is created in the database.

Test Case 3

Test Case #	IT3	Test Name	Display Recommendations
System	RoomieLah	Sub-System	Matching Algorithm
Designed by	QA Team	Design Date	7 April 2021
Executed by	QA Team	Execution Date	8 April 2021
Description	The matching algorithm deployed on a Flask backend should be integrated with the view recommendations screen.		

Pre-conditions	1. User is logged into RoomieLah. 2. Flask backend for the matching algorithm has been deployed and is accessible by the clients

Step #	Action	Expected System Response
1.	The user should navigate to the recommendations tab. The user continuously swipes left or right to exhaust the full list of recommended profiles.	The system should correctly display the details about the users recommended by the algorithm. The username and details should match the details printed out in the console of the server.
2.	User clicks on the profile card to view more information about the profile recommended.	The system displays information like hobbies, interests, personal details etc fetched from the user's collection in Firestore. The displayed details should match the data stored in the document for the user in users' collection on Firestore.

Post-conditions	Users can view dynamically updated recommendations given by Matching algorithm module in the backend

Test Case 4

Test Case #	IT4	Test Name	Real time Chatting
System	RoomieLah	Sub-System	Chat Streamer
Designed by	QA Team	Design Date	7 April 2021
Executed by	QA Team	Execution Date	8 April 2021
Description	Matched users should be able to chat with each other and chats should be updated in real time.		

Pre-conditions	1. 2 registered users have logged into the system 2. The users have matched with each other. 3. Backend application server has been deployed and is listening to client requests.

Step #	Action	Expected System Response
1	Both the users navigate to the Chats tab on the bottom navigation bar.	System displays the list of matches, fetched from the matches collection in Firestore. The displayed usernames should match the respective documents from the matches collection on Firestore.
2.	Each user selects the other from the list of matches and starts messaging to one another.	Each user should be able to chat in real time without loss of messages or occurrence of any other server error.

Post-conditions	Users are able to chat with one another in real time

Test Case 5

Test Case #	IT5	Test Name	Profile Integration
System	RoomieLah	Sub-System	Stream Manager, Profile Controller
Designed by	QA Team	Design Date	7 April 2021
Executed by	QA Team	Execution Date	8 April 2021
Description	When a user edits his/her profile details then the changes should be reflected in the database and the recommended profiles.		

Pre-conditions	1. User is logged into RoomieLah. 2. Backend application server has been deployed and is listening to client requests

Step #	Action	Expected System Response
1	The user navigates to the edit tab in the bottom navigation bar	System displays the current user details fetched from the users collection in the firestore. The displayed details should match the data entries in the user document in the collection.
2.	The user keeps editing every field displayed one by one	Each and every field displayed on the page should change in the required document in the users collection on Firestore.
3	User navigates to the recommendation tab on the nav bar.	System displays a new set of profiles based on the profile details updated previously. One should be able to verify change in preferences easily by viewing the recommended profiles.

Post-conditions	Users are able to update their profiles and view dynamic recommendations from the system.

System Testing

System testing is the process of testing all the integrated hardware and software components of our system to verify that the application meets its specified requirements.

Test Case 1

Test Case #	ST1	Test Name	Navigation Test
System	RoomieLah	Sub-System	Navigation Test
Designed by		Design Date	7 April 2021
Executed by		Execution Date	8 April 2021
Description	The user must be able to navigate across and use all the features designed.		

Pre-conditions	1. The user is logged in to RoomieLah using his credentials 2. The physical device being used is connected to a stable internet connection

Step #	Action	Expected System Response
1	View Recommended Profiles	Swipable cards of all the recommended users are displayed to the user.
2	View all chat tiles	All matches of the user (new and old) are displayed in a list format.
3	Chat with a matched user	The user is able to send and receive text messages from the selected user.
4	Edit Profile details/Preferences	The user is able to view and edit their own profile details - Name, University, Profile Picture, etc and their preferences of the quality/habits of their roommates.

Post-conditions	The complete system is able to interact based on the user's action.

Test Case 2

Test Case #	ST2	Test Name	Installation Test
System	RoomieLah	Sub-System	Installation Test
Designed by		Design Date	7 April 2021
Executed by		Execution Date	8 April 2021
Description	The user must be able to install the mobile application of their mobile phone.		

Pre-conditions	2. The user has a mobile phone 3. The user's mobile phones has a stable internet connection (mobile data or WiFi)

Step #	Action	Expected System Response
1	Installation of the RoomieLah application	The application is successfully installed and launched to the Login Screen on the user's mobile phone

Post-conditions	The RoomieLah mobile application is successfully installed and is ready to use on any mobile device

Test Case 3

Test Case #	ST3	Test Name	Stress Test
System	RoomieLah	Sub-System	Stress Test
Designed by		Design Date	7 April 2021
Executed by		Execution Date	8 April 2021
Description	Multiple users must be able to use RoomieLah parallely without any problem or delay.		

Pre-conditions	3. All mobile devices must have a stable internet connection

Step #	Action	Expected System Response
1.	All users should operate the application and try to navigate through all the pages simultaneously.	The application should work fast, without any delay and smoothly without crashing. The users must be able to use all features of the application with the same ease as if only one user was using it.

Post-conditions	The RoomieLah mobile application is successfully able to handle the stress of multiple users

Test Case 4

Test Case #	ST4	Test Name	Scalable DB Test
System	RoomieLah	Sub-System	Scalable DB Test
Designed by		Design Date	7 April 2021
Executed by		Execution Date	8 April 2021
Description	Multiple users must be able to use RoomieLah Database parallely without any problem or delay.		

Pre-conditions	4. All mobile devices must have a stable internet connection

Step #	Action	Expected System Response
1	All 84 users try logging in to the mobile application simultaneously.	All the users must be logged in without any delay or crashing of the mobile application.
2.	All 84 users should operate the application and try to edit their preferences or profile data parallely	The database should update the preferences for all the users simultaneously and process these preferences to show the recommended groups. The update to Firebase should be smooth and the server should not crash.

Post-conditions	The RoomieLah mobile application database is scalable for multiple users interacting with the database simultaneously.

Test Case 5

Test Case #	ST5	Test Name	Security Test
System	RoomieLah	Sub-System	Security Test
Designed by		Design Date	7 April 2021
Executed by		Execution Date	8 April 2021
Description	Any user/hacker must not be able to enter the application or the database in an unauthorized manner		

Pre-conditions	3. A script to bypass RoomieLah's security firewall is written

Step #	Action	Expected System Response
1	The script is run in an attempt to bypass the security firewall.	The user must not be able to login through the script and must be redirected to the home page. If the user is already logged in, the user must be shown a warning and be logged out of the application.

Post-conditions	The RoomieLah mobile application database is well protected from any unauthorized breaches.