

DESIGN USE CASES



Student Off-Campus Housing App: Unilink

Company

XOD (Extraordinary OO Designers) Inc.

Team Members

Zezhen Xu
Weicong Dong
Dacheng Li
Etsu Nakahara
Guangzheng Zhang
Yijiu Zhong
Chunan Huang
Wenxuan Huang
Shijie Ji
Wenshuo Li

Project Manager
Database Specialist
Algorithm Specialist
Senior System Analyst
Software Development Lead
Database Specialist
Business Analyst
Software Architect
Quality Assurance Lead
User Interface Specialist

Contents

Legend	3
<u>Account</u>	4
A1: User Sign-Up	4
A2: User Sign-In	6
A3: User Logout	8
A4: Password Recovery	9
A5: Change Password	11
A6: Edit Personal Profile	13
A7: View User's Profile	14
<u>Housing</u>	15
H1: Create New Housing Post	15
H2: View Housing Post	16
H3: Edit My Housing Post	17
H4: Delete My Housing Posts	18
H5: Add Housing Posts to My Favorite List	19
H6: View Favorite List	20
H7: Remove Post from Favorites	21
H8: Sort Housing Posts	22
H9: Set Filter	23
H10: Search by Keyword	24
H11: View Recommended Houses List	25
<u>Roommate</u>	26
R1: Set Preference	26
R2: Set Visibility to Others	28
R3: View Recommended Roommates List	30
R4: View Roommate Post	31
R5: Add Users to Your Roommate List	32
<u>Chat</u>	33
C1: Start a New Chat	33
C2: Open Existing Chat	34
C3: Clear Existing Chat	35

Legend

Priority	High, Medium, Low
Status/Test Status	Planned, In-Progress, Implemented, Canceled
Viewer	Users who is looking for a housing or roommate
Poster	Users who post housing information
Favorite	A list of post that viewer selects from all the post that attracts him or her.
Chat	A message box that two users used to communicate.

Account

A1: User Sign-Up back to top	
Priority	High
Status	Implemented
Description	The user creates a new account using email address and password or Google account.
User Goal	The user wants to create a new account for the application.
Desired Outcome	The user has created an account that they can log in with their email and password or Google account.
Actor	Poster/Viewer
Dependent Use Cases	N/A
Requirements	SR A1
Pre-conditions	The user downloads and opens the application but does not have an account.
Post-conditions	The user has created a new account associated with email.
Trigger	The user wants to create an account to use this application.
Workflow	<ol style="list-style-type: none"> 1. The frontend shall render starting page described by activity_authentication.xml. 2. The user shall click on "Sign Up" button. 3. The frontend shall render "Sign-up" page described by activity_authentication.xml. 4. The user shall input their name, email, password in the corresponding text boxes and click the "Sign-up" button. 5. The backend shall verify that the email is a valid email and that the password is at least 6 characters long. 6. The backend shall push account data to firebase in AuthenticationActivity.java. 7. The frontend shall render the "My Profile" page described by activity_initiate_profile.xml. 8. The user shall set their avatar, name, gender, year of graduation and description and click SAVE button. 9. The backend shall push profile data to firebase in AuthenticationActivity.java. 10. The frontend shall navigate the user to Housing page described by fragment_housing.xml.

Alternate Workflow	<p>Use Google account:</p> <ol style="list-style-type: none"> 1. The frontend shall render starting page described by activity_authentication.xml. 2. The user shall click on the “Connect” button. 3. The frontend shall render a window to let user select or enter their Google account. 4. The user shall select or enter their Google account. 5. The backend shall verify and push the account data to firebase described by AuthenticationActivity.java. 6. The backend shall render the “My Profile” page described by activity_initiate_profile.xml. 7. The user shall set their avatar, name, gender, year of graduation and description and click SAVE button. 8. The backend shall push profile data to firebase in AuthenticationActivity.java. 9. The frontend shall navigate the user to Housing page described by fragment_housing.xml. y <p>Empty Name:</p> <ol style="list-style-type: none"> 1. The user shall leave name field blank. 2. The backend shall verify that the name field is empty. 3. The frontend shall render a window to notify so. <p>Email already taken:</p> <ol style="list-style-type: none"> 1. The user shall enter an occupied email. 2. The backend shall verify that the email format is invalid or duplicate in Firebase. 3. The frontend shall render a window to notify so. <p>Invalid Email:</p> <ol style="list-style-type: none"> 1. The user shall enter an invalid or occupied email. 2. The backend shall verify that the email format is invalid or duplicate in firebase. 3. The frontend shall render a window to notify so. <p>Non-matching Passwords:</p> <ol style="list-style-type: none"> 1. The user shall enter different passwords in two password fields. 2. The backend shall verify that the two fields do not match. 3. The frontend shall render a window to notify so.
---------------------------	---

A2: User Sign-In back to top	
Priority	High
Status	Implemented
Description	The user logs into their account using their email and password or Google account.
User Goal	The user wants to log in to the application with their account.
Desired Outcome	The user is logged into the application using their account.
Actor	Poster/Viewer
Dependent Use Cases	DUC A1
Requirements	SR A2
Pre-conditions	The user has an account and is on the start page.
Post-conditions	The user can successfully login their account and will be brought to the housing page of the application.
Trigger	The user wants to log-in to their account.
Workflow	<ol style="list-style-type: none"> 1. The frontend shall render starting page described by activity_authentication.xml. 2. The user shall click "Sign In" button. 3. The frontend shall render Sign-in page described by activity_authentication.xml. 4. The user shall enter their email and password and click "Sign In" button. 5. The backend shall verify the email and password data to account data in Firebase in AuthenticationActivity.java. 6. The system shall render Housing page described by fragment_housing.xml.
Alternate Workflow	<p>Google sign in:</p> <ol style="list-style-type: none"> 1. The frontend shall render starting page described by activity_authentication.xml. 2. The user shall click on the "Connect" button. 3. The frontend shall render a window to let user select or enter their Google account. 4. The user shall select or enter their Google account. 5. The backend shall verify and push the account data to firebase described by AuthenticationActivity.java. 6. The frontend shall render Housing page using fragment_housing.xml. <p>Invalid Email Format:</p> <ol style="list-style-type: none"> 1. The user shall enter an invalid email. 2. The backend shall verify that the email format is invalid. 3. The frontend shall render a window to notify so.

	<p>Non-existing Account:</p> <ol style="list-style-type: none"> 1. The user shall enter an email and a password which have not been registered. 2. The backend shall verify that there is no matching account data in firebase in AuthenticationActivity.java. 3. The frontend shall render a window to notify so. <p>Too Short Password:</p> <ol style="list-style-type: none"> 1. The user shall enter a password that is less than 6 characters. 2. The backend shall verify that the string length is less than 6. 3. The frontend shall render a window to notify so. <p>Incorrect Password:</p> <ol style="list-style-type: none"> 1. The user shall enter an incorrect password that is longer than 6 characters. 2. The backend shall verify that the password data in firebase does not match in AuthenticationActivity.java. 3. The frontend shall render a window to notify so.
--	---

A3: User Logout		back to top
Priority	High	
Status	Implemented	
Description	The user is able to log out of their account.	
User Goal	The user wants to be logged out of the application.	
Desired Outcome	The user is logged out of their account.	
Actor	Poster/Viewer	
Dependent Use Cases	DUC A1, A2	
Requirements	SR A3	
Pre-conditions	The user is logged in to their account.	
Post-conditions	The user is logged out from their account.	
Trigger	The user wants to log out from the application.	
Workflow	<ol style="list-style-type: none"> 1. The user shall click "Me" button. 2. The frontend shall render "Me" page described by fragment_me.xml. 3. The user shall click the "Log out" button. 4. The backend shall log the user out in MeFragment.java. 5. The frontend shall render the starting page described by activity_authentication.xml. 	
Alternate Workflow	N/A	

A4: Password Recovery back to top	
Priority	Medium
Status	Implemented
Description	The user can recover password upon forgetting password.
User Goal	The user wants to reset their password and logs into app using the new password.
Desired Outcome	The user changes their password.
Actor	Poster/Viewer
Dependent Use Cases	DUC A1, A2
Requirements	SR A4
Pre-conditions	The user is on the Sign-in page or logged in to their account and on the "Change Password" page.
Post-conditions	The user's password is reseted.
Trigger	The user forgets their password and wants to recover the account.
Workflow	<p>Recover password from Sign-in page:</p> <ol style="list-style-type: none"> 1. The user shall enter their email and click the "Forget Password?" button in "activity_authentication.xml". 2. The frontend shall verify the email format. 3. The backend controller shall verify the existence of the entered account in the database. 4. The backend shall send a password recovery email to the user. 5. The frontend shall prompt "Your password reset link is sent to your email" in "activity_authentication.xml". 6. The user shall click on the link in their email and reset their password. 7. The backend shall reset the password for the user. <p>Recover password from "Change Password" page:</p> <ol style="list-style-type: none"> 1. The user shall click the "Forget Password?" button in "activity_my_change_password.xml". 2. The backend shall send a password recovery email to the user. 3. The frontend shall prompt "Your password reset link is sent to your email" in "activity_my_change_password.xml". 4. The user shall click on the link in their email and reset their password. 5. The backend controller shall reset the password for the user in the database.
Alternate Workflow	<p>Invalid Email Format:</p> <ol style="list-style-type: none"> 1. The user enters a email with wrong format. 2. The frontend shall prompt "Please enter a valid Email!"

	<p>Account does not exist:</p> <ol style="list-style-type: none">1. The user enters a non-existing account.2. The backend controller shall verify the existence of the entered account in the database.3. The frontend shall prompt "Email is not found."
--	---

A5: Change Password back to top	
Priority	Medium
Status	Implemented
Description	The user can change their password on “Me” page.
User Goal	The user wants to reset their password.
Desired Outcome	The password has been reset.
Actor	Poster/Viewer
Dependent Use Cases	DUC A1, A2
Requirements	SR A5
Pre-conditions	The user is logged in to their account and on the “Me” page.
Post-conditions	The user can login with a new password.
Trigger	The user wants to change their password.
Workflow	<ol style="list-style-type: none"> 1. The user shall click the “Change Password” button in “fragment_me.xml”. 2. The frontend shall navigate the user to the “Change Password” page and present the user a form to enter the old, new password and confirm password in “activity_my_change_password.xml”. 3. The user shall enter their old password, new password and confirm password. 4. The front end shall validate the new password is valid and matches. 5. The backend controller shall verify the old password from the database. 6. The system shall change the password and prompt that “Your Password has been Changed!” 7. The system shall navigate the user to “activity_authentication.xml”.
Alternate Workflow	<p>Wrong Old Password:</p> <ol style="list-style-type: none"> 1. The user shall enter a wrong old password. 2. The system shall stay on “fragment_me.xml” page and prompt that “The password is invalid or the user does not have a password”. <p>New Password Too Short:</p> <ol style="list-style-type: none"> 1. The user shall enter a new password that is less than 6 characters. 2. The system shall stay on “fragment_me.xml” page and prompt that “The given password is invalid: [Password should be at least 6 characters]”. <p>Confirm Passwords Not Match:</p> <ol style="list-style-type: none"> 1. The user shall enter a confirm password that is different to new

	<p>password entered.</p> <p>2. The system shall stay on "fragment_me.xml" page and prompt that "Please make sure your new password is the same as your confirm password".</p> <p>The user logs application using Google sign-in:</p> <p>1. The user shall click "Change Password" button.</p> <p>2. The system shall prompt " You can only change your password if you sign up using email, please change your Google password if you use Google login!" in "fragment_me.xml".</p>
--	--

A6: Edit Personal Profile back to top	
Priority	High
Status	Implemented
Description	The user will be able to edit their personal information.
User Goal	The user wants to be able to edit and display their profile information to other users.
Desired Outcome	The user will have edited their profile information.
Actor	Poster/Viewer
Dependent Use Cases	DUC A1, A2
Requirements	SR A6
Pre-conditions	The user is logged in and on "Me" page.
Post-conditions	The user's profile is edited.
Trigger	The user wishes to edit their personal information
Workflow	<ol style="list-style-type: none"> 1. The user shall click the "Edit Profile" button. 2. The frontend shall render the "activity_my_change_profile.xml" page. 3. The user shall edit the profile. 4. The user shall click "SAVE" button. 5. The backend shall update information using the "User.java" model and save to the database in "ChangeProfileActivity.java". 6. The frontend shall navigate to "fragment_me.xml" page.
Alternate Workflow	Empty "Name" field: <ol style="list-style-type: none"> 1. The user does not fill the "Name" field. 2. The frontend shall prompt "Please enter a valid name!"

A7: View User's Profile back to top	
Priority	High
Status	Implemented
Description	View any user's personal profile page.
User Goal	The user can view any user's profile.
Desired Outcome	The user views any user's profiles.
Actor	Poster/Viewer
Dependent Use Cases	DUC A1, A2, A6
Requirements	SR A7
Pre-conditions	The user is logged in and is on any page with a user's avatar.
Post-conditions	The profile of the user whose avatar is clicked is rendered.
Trigger	The user wants to view any user's profile
Workflow	<ol style="list-style-type: none"> 1.The user shall click on the avatar of the user whose profile the viewer wants to check. 2. The backend shall retrieve other user's profile data from database and save to "User.java" model. 3.The frontend shall render the "activity_profile.xml" page with the "User.java" model.
Alternate Workflow	N/A

Housing

H1: Create New Housing Post back to top	
Priority	High
Status	Implemented
Description	The user can add their own housing post to the app.
User Goal	The user wants to add their own housing posts containing their releasing information.
Desired Outcome	A new housing post containing the user house's information is created.
Actor	Poster
Dependent Use Cases	DUC A1, A2
Requirements	SR H1
Pre-conditions	The user is logged in and is on "Housing" page.
Post-conditions	A new housing post is added to the app.
Trigger	The user wants to create a new housing post.
Workflow	<ol style="list-style-type: none"> 1. The frontend shall render the housing page described by frgment_housing.xml. 2. The user shall click on the "+" button. 3. The system shall navigate to the "Add Post" page. 4. The frontend shall render a form described by activity_add_post.xml. 5. The user shall enter all information of their house. 6. The user shall click "SUBMIT" button. 7. The backend shall check every information is filled. 8. The backend shall push the added post to firebase in AddPost.java. 9. The system shall navigate back to "Housing" page. 10. The frontend shall render the housing page described by frgment_housing.xml.
Alternate Workflow	<p>Incomplete Post Information:</p> <ol style="list-style-type: none"> 1. The user shall leave at least one required fields blank in the form and click "SUBMIT". 2. The backend shall find out that not every information is filled in AddPost.java. 3. The frontend shall render a toast to notify the user in activity_add_post.xml.

H2: View Housing Post back to top	
Priority	High
Status	Implemented
Description	The user can view any user's housing post(s).
User Goal	The user wants to review existing housing post(s).
Desired Outcome	The user can view existing post(s).
Actor	Poster/Viewer
Dependent Use Cases	DUC A1,A2, H1
Requirements	SR H2
Pre-conditions	The user is logged in and on "Housing" page.
Post-conditions	The certain housing post page is rendered.
Trigger	The user wants to view housing post(s).
Workflow	<ol style="list-style-type: none"> 1. The backend shall pull HousePost objects from Firebase using "HousingFragment.java". 2. The frontend shall render a list of housing posts using "fragment_housing.xml" 3. The user shall click on the post they want to view. 4. The backend shall get the House object from Firebase using "HousePostActivity.java" 5. The frontend shall navigate to the corresponding housing post page using "activity_house_post.xml".
Alternate Workflow	<p>View posts from My Favorite:</p> <ol style="list-style-type: none"> 1. The user shall click on "My Favorite" button at "Me" page. 2. The frontend shall navigate to "My Favorite" page using "activity_my_favorite.xml". 3. The user shall click on the post they want to view. 4. The frontend shall navigate the corresponding housing post page. <p>View posts from My Post:</p> <ol style="list-style-type: none"> 1. The user shall click on "My Post" button at "Me" page. 2. The frontend shall navigate to "My Post" page using "activity_my_post.xml". 3. The user shall click on the post they want to view. 4. The frontend shall navigate to the corresponding housing post page using "activity_house_post.xml".

H3: Edit My Housing Post back to top	
Priority	High
Status	Implemented
Description	The user can edit their housing posts.
User Goal	The user wants to update their posts
Desired Outcome	The user can update the information of their posts.
Actor	Poster
Dependent Use Cases	DUC A1, A2, H1
Requirements	SR H3
Pre-conditions	1. The user is logged in and is on "My Post" page. 2. The user has created a housing post.
Post-conditions	The user's post is updated accordingly to the edit.
Trigger	The user wants to update their posts or fix some errors in their posts.
Workflow	<ol style="list-style-type: none"> 1. The user shall click the edit icon at upper-right screen. 2. The frontend shall render an arrow at right of the post as described at activity_my_post.xml. 3. The user shall click on the arrow. 4. The system shall navigate the user to "Edit Post" page. 5. The frontend shall render a form as described in activity_edit_post.xml 6. The user shall edit the textural info or the picture of the post by changing the texts in the text fields or by delete or uploading a new picture. 7. The user shall click on "SUBMIT" button. 8. The backend shall check every information is filled. 9. The backend shall push the added post to firebase in EditPost.java. 10. The system shall navigate back to "My Post" page.
Alternate Workflow	<p>Incomplete Post Information:</p> <ol style="list-style-type: none"> 1. The user shall leave at least one required fields blank in the form and click "SUBMIT". 2. The backend shall find out that not every information is filled in EditPost.java. 3. The frontend shall render a toast to notify the user in activity_edit_post.xml.

H4: Delete My Housing Posts back to top	
Priority	High
Status	Implemented
Description	The user can delete their post.
User Goal	The user wants to delete their post to stop promoting a house.
Desired Outcome	Housing post is deleted.
Actor	Poster
Dependent Use Cases	DUC A1, A2, H1, H2
Requirements	SR H4
Pre-conditions	The user is on "My Post" page and there is at least one user post.
Post-conditions	Post is deleted and can no longer be seen by other users.
Trigger	<ol style="list-style-type: none"> 1. The user reached agreement with a roommate or rentee and wants to stop promoting. 2. The user gives up all current activities and wants to stop receiving contact from other users.
Workflow	<ol style="list-style-type: none"> 1. The user shall click the edit icon at the upper-right of "activity_my_post.xml" page. 2. The frontend shall render a red symbol button at the left of each post. 3. The user shall click the red symbol button. 4. The frontend shall render a window prompt asking the user that "Are you sure you want to delete this post? The action cannot be reversed." 5. The user shall click "DELETE" button. 6. The backend shall delete the corresponding "House.java" model, and deleted that post's post_id from the my_house_posts list in the "User.java" model. 7. The frontend shall refresh "activity_my_post.xml" page.
Alternate Workflow	<p>Cancel deletion:</p> <ol style="list-style-type: none"> 1. The user shall click the edit icon at the upper-right of "activity_my_post.xml" page. 2. The frontend shall render a red symbol button at left of each post. 3. The user shall click the red symbol button. 4. The frontend shall render a window prompt asking "This post is to be deleted. Are you sure? " 5. The user shall click "CANCEL" button. 6. The frontend shall remove the window prompt and stay at "activity_my_post.xml" page without making any change.

H5: Add Housing Posts to My Favorite List back to top	
Priority	Medium
Status	Implemented
Description	The user can add housing posts that they are interested into “My Favorite” so that they can quick access the posts later.
User Goal	The user wants to select and keep control of a personal list of posts that they will be focus on.
Desired Outcome	The user add a housing post to a favorite list.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1, H2
Requirements	SR H5
Pre-conditions	The user is properly logged in and on “Housing” page where some posts exists. Or the user is on a particular post page.
Post-conditions	Favorite icon at upper right of post is red and the user can find this post at favorite list page.
Trigger	The user is interested in particular posts.
Workflow	<ol style="list-style-type: none"> 1. The user shall click the favorite icon at the upper right of “activity_house_post.xml” page. 2. The backend shall store the post’s post_id into the favorite_houses list in the “User.java” model. 3. The frontend shall render the favorite icon in red color.
Alternate Workflow	N/A

H6: View Favorite List back to top	
Priority	Medium
Status	Implemented
Description	The user can see a list of the favorite posts they saved previously.
User Goal	The user wants to see their favorite posts.
Desired Outcome	The user can see a list of their favorite posts.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1, H5
Requirements	SR H6
Pre-conditions	1. The user is logged in. 2. The user is on "Me" page.
Post-conditions	The user can see a list of their favorite posts.
Trigger	The user wants to see their favorite posts.
Workflow	1. The user shall click on the "My Favorite" button on "fragment_me.xml" page. 2. The backend shall fetch the favorite_houses list from "User.java" model, and the post information of each posts in the favorite_houses list from "House.java" model. 3. The frontend shall render "activity_my_favorite.xml" page and a list of user's favorite houses in the page.
Alternate Workflow	N/A

H7: Remove Post from Favorites back to top	
Priority	Medium
Status	Implemented
Description	The user can delete a housing post from their favorite list.
User Goal	The user wants to remove a housing post from favorite list.
Desired Outcome	The user deletes a housing post from favorite list.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1, H2, H5, H6
Requirements	SR H7
Pre-conditions	1. The user is on My Favorite page or Housing page or particular Post page. 2. The user has favorited the certain housing post.
Post-conditions	Deleted post disappears from user's favorite list and the Favorite icon at upper right of post is empty.
Trigger	The user no longer wants to focus on a housing post.
Workflow	1. The user shall click the favorite icon on a post. 2. The backend shall remove the housing post from the user's favorite list in Firebase using "HousePostAdaptor.java" or "HousePostActivity.java" or "MyFavoriteAdaptor.java" based on the pre-condition. 3. The frontend shall remove the post from favorite list and render a favorite list page without the removed post using "fragment_housing.xml" or "activity_house_post.xml" or "house_list_item_favorite.xml" based on the pre-condition.
Alternate Workflow	N/A

H8: Sort Housing Posts back to top	
Priority	Medium
Status	Implemented
Description	The user can sort the post they see according to specific criteria.
User Goal	The user wants to see posts in sorted order of specific feature.
Desired Outcome	The user sees posts in sorted order of specific feature.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1, H2
Requirements	SR H8
Pre-conditions	The user is on "Housing" page.
Post-conditions	The posts is sorted in specific order.
Trigger	The user wants to see the housing posts in a sorted order.
Workflow	<ol style="list-style-type: none"> 1. The user shall click the sort button at upper-right of the page. 2. The frontend shall render a list of sorting criteria "fragment_housing.xml". 3. The user shall choose "Price: low to high" or "Price: high to low" or "Term: long to short" or "Term: short to long" in the rendered list of criteria. 4. The backend shall apply the sorting criteria using "HousingFragment.java". 5. The frontend shall display posts based on the chosen criteria using "fragment_housing.xml".
Alternate Workflow	N/A

H9: Set Filter back to top	
Priority	Medium
Status	Canceled
Description	The user can edit filters to see posts that satisfy certain requirements
User Goal	The user want to see posts that satisfy certain requirements
Desired Outcome	The user can edit the filter to see posts that meets requirements
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1
Requirements	SR H9
Pre-conditions	The user is logged in and on "Housing" page
Post-conditions	The user successfully add/edit the filter setting
Trigger	The user only want to see posts that satisfy their basic needs
Workflow	<ol style="list-style-type: none"> 1. The user shall click the filter button. 2. The backend shall get the user's current filter. 3. The frontend shall render filter page with current settings. 4. The user shall edit the filter and click "submit". 5. The backend shall update user's filter information. 6. The backend shall filt the posts according to the user's preference. 7. The frontend shall render a list of house posts.
Alternate Workflow	Clear filter: <ol style="list-style-type: none"> 1. The user shall click the filter button. 2. The frontend shall render filter page. 3. The user shall click the "clear" button in the filter. 4. The backend shall delete the user's filter and save the empty filter. 5. The backend shall update user's filter information. 6. The frontend shall render a house posts list with no filter.

H10: Search by Keyword back to top	
Priority	Medium
Status	Canceled
Description	The viewer can search information by entering keyword.
User Goal	The user wants to search post by keyword.
Desired Outcome	The viewer will be able to view information about certain keyword.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1
Requirements	SR H10
Pre-conditions	The user is logged in and on "Housing" page.
Post-conditions	The searched results about the certain keyword will be listed.
Trigger	The viewer wants to view posts that contain certain keyword.
Workflow	<ol style="list-style-type: none"> 1. The user shall click on the search bar on top. 2. The user shall enter the keyword they wants to search and click on the search button. 3. The backend shall search for the keyword in database. 4. The frontend shall render posts containing the keyword on "Housing" page.
Alternate Workflow	N/A

H11: View Recommended Houses List back to top	
Priority	Low
Status	Canceled
Description	The users can see a list of recommend housing provided by the system.
User Goal	The user want to see recommended houses.
Desired Outcome	Users can get recommended houses list that they may like.
Actor	Viewer
Dependent Use Cases	DUC A1, A2, H1
Requirements	SR H8
Pre-conditions	<ol style="list-style-type: none"> 1. The user is logged in. 2. The user fills out their preference about house.
Post-conditions	A list of recommended houses is shown in the app
Trigger	Users want to see a list of recommended houses they may like, rather than spending time in looking for houses themselves.
Workflow	<ol style="list-style-type: none"> 1. The user shall click on the "recommended houses" button. 2. The backend shall get the user preference information. 3. The backend shall use the user information and the algorithm to find the houses that the user may like out of all the posts in the firebase. 4. The frontend shall render the list of houses that backend has found.
Alternate Workflow	<p>Not enough information:</p> <ol style="list-style-type: none"> 1. The user shall click on the "recommended houses" button. 2. The backend shall get the user preference information and find the preference is not set. 3. The frontend shall render a prompt that "Profile information not enough for this functionality", and provide a button "fill out information" which can redirect the user to the page of "editing profile".

Roommate

R1: Set Preference back to top	
Priority	High
Status	Implemented
Description	The user can set personal preference.
User Goal	The user wants to find their favorite roommates.
Desired Outcome	The system can get preference information of the user.
Actor	Viewer/Poster
Dependent Use Cases	DUC A1, A2
Requirements	SR R1
Pre-conditions	The user is logged in and is on the "Me" Page.
Post-conditions	System gets all necessary information to perform an evaluation on suitable roommates.
Trigger	The user wants to find suitable roommates.
Workflow	<ol style="list-style-type: none"> 1. The user shall click "My Preference" button on the middle of the page. 2. The frontend shall navigate the user to "My Preference" page using "activity_my_preference.xml". 3. The user shall select all information on their preference and click Save button. 4. The backend shall process data that the user has entered and create a Preference object using "Preference.java" and logic in "ChangePreferenceActivity.java". 5. The backend should send the Preference object to the Firebase. 6. The frontend shall navigate the user back to the Me page.
Alternate Workflow	<p>User hasn't finished the preference form:</p> <ol style="list-style-type: none"> 1. The user shall click on the Roommate button on the navigation bar. 2. The frontend shall navigate the user to Roommate page. 3. The backend shall detect that the user hasn't finished the preference form using "RoommateFragment.java". 4. The frontend shall render a prompt saying that "Without preference information, we are unable to provide you with the best possible roommate and you will be invisible to other users". 5. The user shall click "SET PREFERENCE" Button.

	<p>6. The frontend shall navigate the user to My Preference page.</p> <p>Empty field(s):</p> <ol style="list-style-type: none"> 1. The user shall click SAVE button while not finishing all required fields in the form. 2. The backend shall detect that the user hasn't finished all the fields using "ChangePreferenceActivity.java". 3. The frontend shall render a prompt saying that "Please finish the survey".
--	---

R2: Set Visibility to Others back to top	
Priority	Medium
Status	Implemented
Description	The user can make their information visible or invisible in other users' roommate page.
User Goal	The user wants other users able or unable to see their information in roommate page.
Desired Outcome	All users can see user who set their status visible in roommate page.
Actor	Viewer/Poster
Dependent Use Cases	DUC A1, A2, R1
Requirements	SR R2
Pre-conditions	The user is logged in and is on Roommate page.
Post-conditions	The switch is on and the user's information appears on other users' Roommate page.
Trigger	The user wants other users to know about them in order to find suitable roommates.
Workflow	<ol style="list-style-type: none"> 1. The backend shall load the user's previous visibility setting. 2. The frontend shall set the "Visible to Others" switch to the corresponding state using RoommateFragment.java. 3. The user shall click the The user shall click the "Visible to Others" switch at upper-right of the page. 4. The frontend shall switch the "Visible to Others" button. 5. The backend shall render the user's information on all other user's Roommate page visible or invisible.
Alternate Workflow	<p>Roommate Preference Not Set 1:</p> <ol style="list-style-type: none"> 1. The user shall click "Visible to Others" button at upper-right of the page. 2. The backend shall get the preference information and find the preference is not set. 3. The frontend shall render a prompt saying that "Without preference information, we are unable to provide you with the best possible roommate and you will be invisible to other users". 4. The user shall click "STILL LOOK AROUND" button. 5. The frontend shall render a roommate list without matching scores. <p>Roommate Preference Not Set 2:</p> <ol style="list-style-type: none"> 1. The user shall click "Visible to Others" button at upper-right of the page.

	<ol style="list-style-type: none">2. The backend shall get the preference information and find the preference is not set.3. The frontend shall render a prompt saying that "Without preference information, we are unable to provide you with the best possible roommate and you will be invisible to other users".4. The user shall click "SET PREFERENCE" button.5. The frontend shall navigate the user to My Preference page.
--	--

R3: View Recommended Roommates List back to top	
Priority	Medium
Status	Implemented
Description	The user can view a list of roommates with matching score.
User Goal	The user wants to find suitable roommates.
Desired Outcome	Users can see a list of roommates with matching scores.
Actor	Viewer/Poster
Dependent DUCs	DUC A1, A2, R1, R2
Requirements	SR R3
Pre-conditions	1. The user is logged in. 2. The user has set preference information.
Post-conditions	A list of roommates is shown with matching scores
Trigger	The user wants to find suitable roommates.
Workflow	1. The user shall click on the "Roommate" button on navigation bar. 2. The frontend shall render the "Roommate" page using "fragment_roommate.xml" 3. The backend shall access the preference data of the user and generate matching score of each other users whose status is set to "visible to others" using "RoommateListAdaptor.java". 4. The frontend shall render the roommate list with matching scores generated by backend on Roommate page.
Alternate Workflow:	N/A

R4: View Roommate Post back to top	
Priority	Medium
Status	Implemented
Description	The user can view details of a roommate post.
User Goal	The user wants to see detailed information of a roommate candidate.
Desired Outcome	The user sees all the detailed information of a roommate candidate.
Actor	Viewer/Poster
Dependent Use Cases	DUC A1, A2, A6, R1, R3
Requirements	SR R4
Pre-conditions	1.The user is on the Roommate page. 2.There is at least one roommate post.
Post-conditions	A page with all the detailed informations for this roommate is rendered to the user.
Trigger	The user is interested in a particular roommate candidate and wants to see more details about him/her.
Workflow	1. The user shall click on a particular roommate post. 2. The backend shall pull a User object from the Firebase using "RoommatePostActivity.java". 3. The frontend shall navigate to the roommate page of the certain roommate.
Alternate Workflow	N/A

R5: Add Users to Your Roommate List back to top	
Priority	Low
Status	Canceled
Description	The user should be able to add other users to their roommates list once both parties agree, and their listing in finding roommates page should be automatically removed.
User Goal	Confirm roommates to "reserve them" and avoid the potential conflict of roommates getting teamed up with others
Desired Outcome	Users can successfully confirm other people as their desired roommates and have them reserved
Actor	Viewer/Poster
Dependent Use Cases	DUC A2
Requirements	SR R5
Pre-conditions	1. The user has an account on the app. 2. The user is logged in.
Post-conditions	The finding for roommates posts of both users deleted
Trigger	Users may want to confirm their roommates so that they cannot be matched or found with others
Workflow	N/A
Alternate Workflow	N/A

Chat

C1: Start a New Chat back to top	
Priority	Medium
Status	Implemented
Description	The viewer can start a new in-app conversation with the poster.
User Goal	The viewer wants to contact the poster of the post they are interested in.
Desired Outcome	The viewer shall be able to type in messages and send to the poster
Actor	Viewer
Dependent Use Cases	DUC A1, A2, A6, H2
Requirements	SR C1
Pre-conditions	The user is logged in and on a particular housing post page or roommate profile page.
Post-conditions	The user is on the chat page and can send a message to the poster.
Trigger	The viewer wants to contact the poster.
Workflow	<ol style="list-style-type: none">1. The user shall click Contact button.2. The frontend shall navigate the user to chat page with the specific poster using "activity_chat.xml"3. The user shall type in the message in the text field.4. The user shall click SEND button to send the message.5. The backend shall a Chat object to the Firebase using "RealtimeDbChatActivity.java".6. The frontend shall render the user's message at right of the screen and notify the other user.
Alternate Workflow	N/A

C2: Open Existing Chat back to top	
Priority	Medium
Status	Implemented
Description	The user should be able to open previous conversations in order to contact the same user again.
User Goal	The user wants to talk to users who they have previously talked to.
Desired Outcome	The user can continue their conversations with the same user.
Actor	Viewer/Poster
Dependent Use Cases	DUC A2, C1
Requirements	SR C2
Pre-conditions	1.The user has logged in. 2.The user has an existing conversation with another user.
Post-conditions	The user is able to see the previous conversations
Trigger	The user wants to contact another user who they have previously talked to.
Workflow	<ol style="list-style-type: none"> 1. The user shall click on the "Chat" button on the navigation bar. 2. The frontend shall navigate the user to Chat page using "activity_chat.xml". 3. The backend shall check that the user has chatted with others using "ChatFragment.java" and pull all ChatListViewHolder object from the Firebase. 4. The frontend shall display these objects.
Alternate Workflow	<ol style="list-style-type: none"> 1. The user shall click Housing button on the navigation bar. 2. The frontend shall navigate the user to Housing page. 3. The user shall click a specific post. 4. The frontend shall navigate the user to Post page of this post. 5. The user shall click on Contact button on the button right. 6. The backend shall pull existing chat data with the certain user from the Firebase using RealtimeDbChatActivity.class. 7. The frontend shall navigate the user to chat page to this specific poster.

C3: Clear Existing Chat back to top	
Priority	Medium
Status	Canceled
Description	The user will be able to clear an existing conversation with other user
User Goal	The user wants to delete unwanted chat messages.
Desired Outcome	The user is able to delete unwanted chat messages.
Actor	Viewer/Poster
Dependent Use Cases	DUC C1, C2
Requirements	SR C3
Pre-conditions	The user is logged in and has existing chats and is on Chat page.
Post-conditions	The deleted chat messages no longer exist.
Trigger	The user wants to delete a finished conversation
Workflow	<ol style="list-style-type: none"> 1. The user shall choose an existing chat with other user. 2. The user shall swipe the chat to the left and click the Delete Button. 3. The backend shall send a request to delete the chat to the Firebase and delete it from the Firebase. 4. The frontend shall render a prompt saying "deleted" and hide the deleted chat.
Alternate Workflow	N/A