ParkShare
Project Documentation

Team 2
Version: 2.0
Date: 2021/12/08

Team Members

Robert, Goss - 180897390
Timothy, Mihet - 180963200
Sourav, Pandya - 170228640
Janelle, Tait - 180447860
Dayton, Talarico - 180783280

Versions

Date	Summary of Changes	Author(s)
Sept 30th	Finished project proposal and started researching implementation	Robert Goss, Timothy Mihet, Sourav Pandya, Janelle Tait, Dayton Talarico
Oct 18th	Finished iteration one	Robert Goss, Timothy Mihet, Sourav Pandya, Janelle Tait, Dayton Talarico
Nov 23th	Implemented database for login	Timothy Mihet, Sourav Pandya
Nov 25th	Implemented my account page and updating users records	Timothy Mihet, Sourav Pandya
Nov 26th	Google Maps added	Robert Goss, Janelle Tait, Dayton Talarico
Nov 29th	Implemented database for posts	Robert Goss, Janelle Tait, Dayton Talarico
Nov 31st	Implemented list views and deletion/adding to database	Robert Goss, Janelle Tait, Dayton Talarico
Dec 8th	Final documentation and revisions of code	Robert Goss, Timothy Mihet, Sourav Pandya, Janelle Tait, Dayton Talarico

Table Of Contents

1. Introduction

ParkShare is an android application that provides users a platform to host or rent parking availabilities. This document describes all the features available in our app and is intended to be a reference for our professor Abdul-Rahman Mawlood-Yunis and states our project's requirements.

1.1 Purpose

Demand for parking is steadily increasing alongside the rising growth rate of vehicle registrations, although supply is not following the same trend. Due to this, parking in heavily populated areas tends to be a significant inconvenience, while many reserved parking spots remain empty for long periods of time. Our app aims to connect drivers in need of parking to hosts of parking vacancies with a mutually beneficial solution.

1.2 Scope

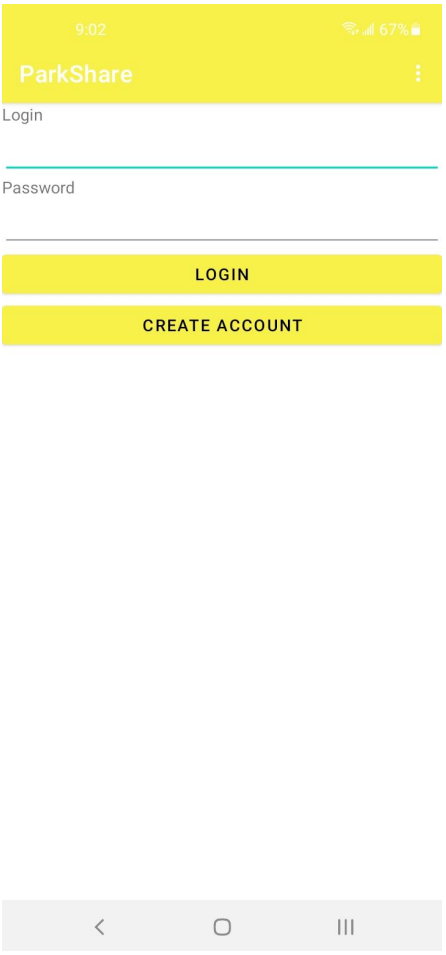
The scope of our app is to create a more convenient, and reliable alternative to parking. Drivers will be able to quickly locate parking availability close to their desired location. Hosts can post their parking vacancies whenever they're available, generating passive income from an otherwise unoccupied spot. Overall this will increase the sustainability of parking and maximise the utilisation of parking space.

1.3 Definitions

- **USER**
  Anyone who is a user of the app with an account they have created
- **POST**
  Parking spot information added in the post activity
- **PIN**
  A marked location on the map at the exact address of a posting
- **HOST**
  A user of the app who has posted one or more parking spaces for rent
- **ACTIVITY**
  Any page in the application where the user interacts with the application

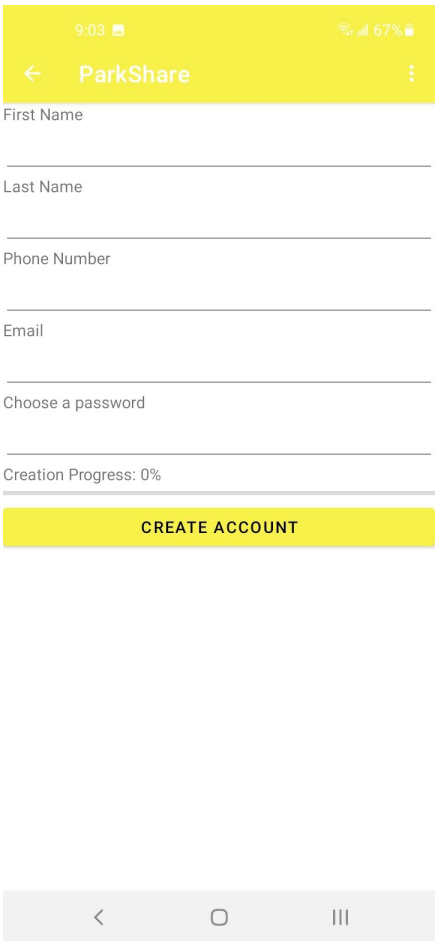
2. User Interfaces

2.1 Login Activity



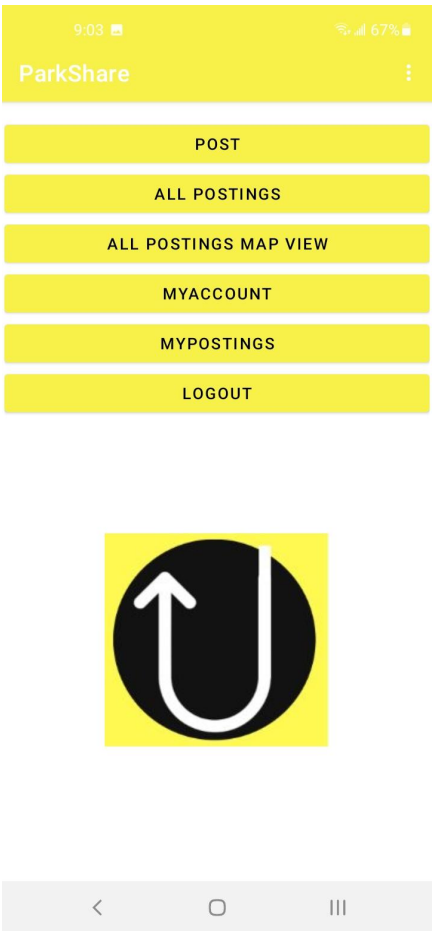
- **User sign-in**
  - Prompts users to enter their email address and password to login assuming the account creation process has been completed.
  - Toolbar includes a help item which displays instructions on how to use the login page.

2.2 Create Account Activity



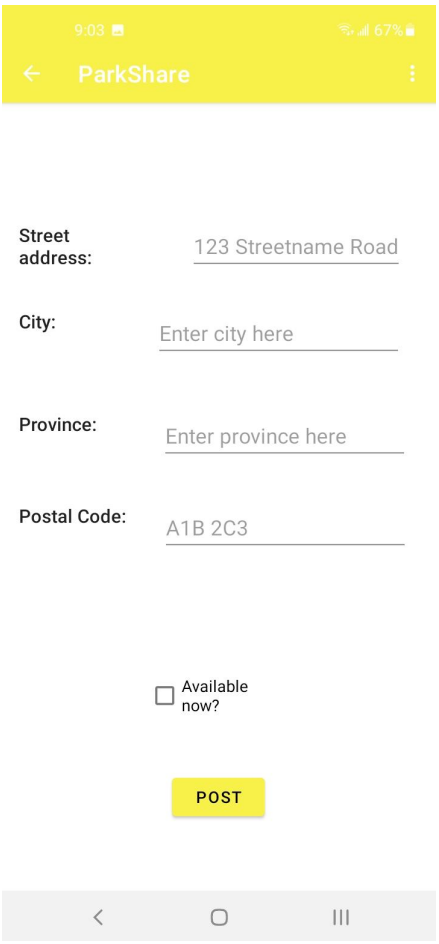
- **User sign-up**
  - Upon joining ParkShare, users must sign up for an account. Users will be prompted to enter their first name, last name, phone number, email, and password.
  - Account creation progress will be displayed and updated as users fill out the listed fields.
  - User information will be stored in the database under the user table as account creation is successful.
  - Toolbar includes a help item which displays instructions on how to use the account creation page.

2.3 Main Activity



- **Main Hub**
    - Initial path to enter all other features offered in ParkShare. Links to Post, All Postings, Map View, MyAccount, and MyPostings activities.
    - Allows users to logout and be brought back to the login page.
    - Toolbar includes a help item which displays instructions on how to use the main page.

2.4 Post Activity

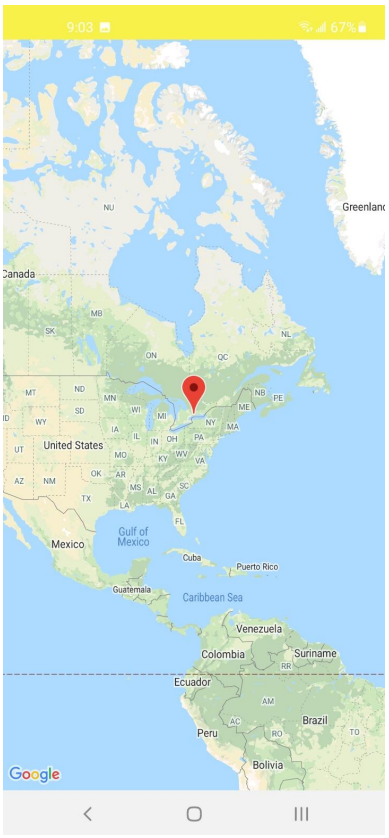


- **Post Parking Spot**
    - A host must enter fields street address, city, province, and postal code in order to create a post.
    - The host has the option to check the “available now?” checkbox in which the parking spot will currently be available and displayed under Current Postings page in the list of posts. Otherwise the post will only appear to the host in the MyPostings page as it is not available to the public.
    - Toolbar includes a help item which displays instructions on how to use the post page.

2.5 All Postings and MyPostings Activities

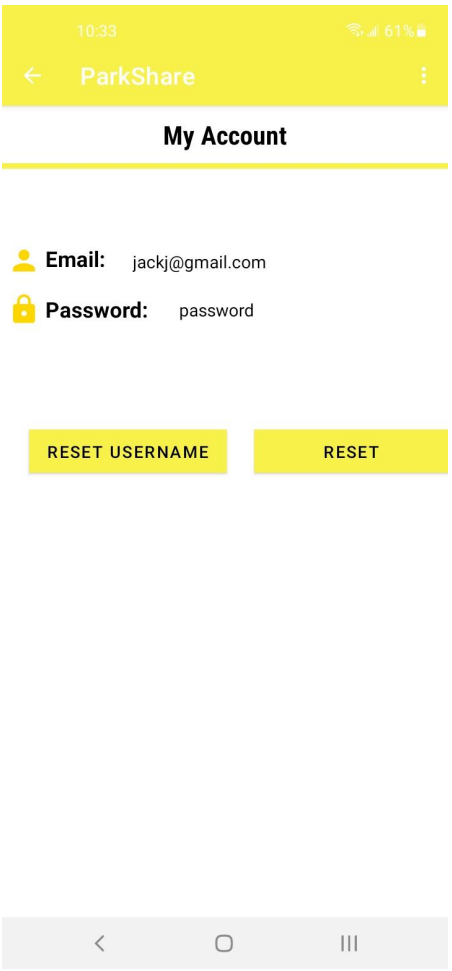
- **All Postings**
    - List view of all currently available parking spots displaying an auto-incremented post id as well as the street address of each spot's location.
    - Clicking on a post will open a dialog that displays more information regarding the parking location including street address, city, province, postal code. This dialog also includes the contact information of the host allowing the user to contact the host in order to discuss the renting of said spot.
    - Toolbar includes a help item which displays instructions on how to use the current postings page
- **MyPostings**
    - List view of all posts hosted by the current user displaying the post id as well as the street address of each spot's location.
    - Clicking on a post will open a dialog that displays more information regarding the parking location including street address, city, province, postal code. This dialog also allows hosts to delete their posts, in which the post will be removed from the database, map view, and both postings list views.
    - Toolbar includes a help item which displays instructions on how to use the MyPostings page

2.6 All Postings Map View Activity



- **Map View**
    - Google maps interface containing the pins of all parking spots.
    - Pins are displayed in green if the parking spot is currently available.
    - Pins are displayed in red if the parking spot is currently unavailable.
    - Clicking on a pin will display the street address of the parking spot.
    - This activity is updated every time it is opened accessing all posts in our database.

2.7 MyAccount Activity



- **Account Information**
    - Displays current user email and password.
    - The “Reset Username” button opens a dialog that allows the user to update their email.
    - The “Reset” button opens a dialog that allows the user to update their password.
    - Toolbar includes a help item which displays instructions on how to use the MyAccount page

3. Overall Description

3.1 User Characteristics

The app has two types of users: those looking to rent out their parking space and those looking for a space to rent. This application is designed for anyone travelling to or currently living in more urban areas where parking spaces are at a premium. Another type of users is someone who has a parking space they own that is going unused for some amount of time and they want to generate some income from the parking space. Users who are renters will be those who need a space for some amount of time. This could be for work, travel or an event they are attending for a short period of time.

3.2 Constraints

The first constraint on our application is time. We were given a strict time limit of 4 months to develop the application with no flexibility for deadlines.

Another constraint on our application is cost. We were given a budget of $0.00 to develop our application. This meant that we could not pay to use a commercial level server to store our app's database and therefore all of the application data is stored locally on the user's device. Our application would be more useful if we were able to pay for a server because different users would then be able to access each other's posts and not only their own.

The last constraint on our application is reliability. Our application must always store and output the correct data exactly as originally entered in by the user. For example, it would be unacceptable for our application to display a posting as being located at 150 Main St. if the original poster had entered the street address as 149 Main St. when they created their posting.

3.3 Assumptions and Dependencies

It is assumed that the users of the application will be looking for either renting out a parking space or renting a parking space they own. It is also assumed that all users will be using some kind of android device since the application is an android app. It is also assumed that the user's device has access to a stable internet connection as this is needed for the application to function (specifically for Google Maps).

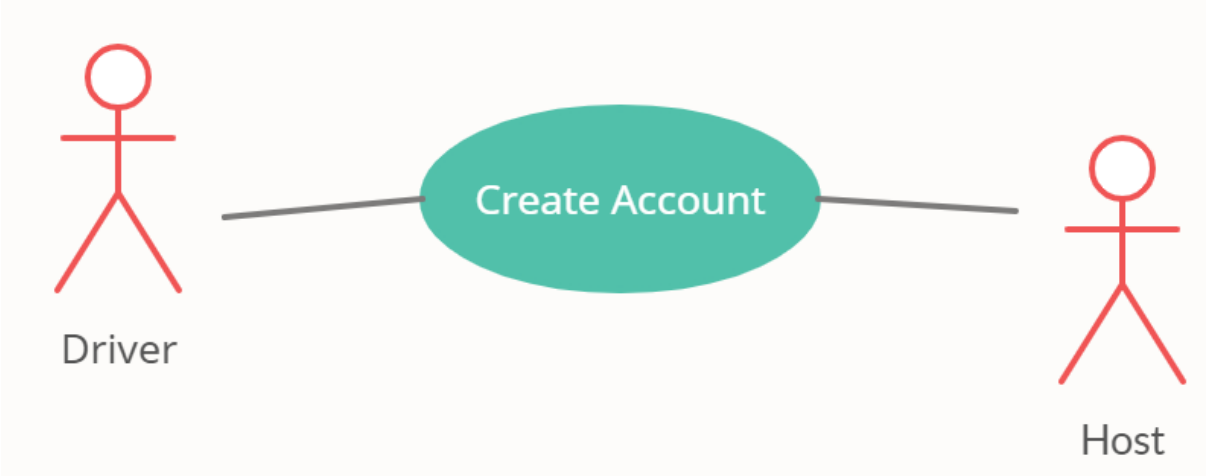
The main dependency outside of what we developed is Google Maps and the underlying Android system. Specifically, Google Maps is dependent upon for the entire functionality of the "all posting map view" page of our application. If something were to change on Google's end, our map page would be non-functional and we would need to re-design the page to comply with the changes.

4. Specific Requirements

4.1 Functional Requirements

Requirement No.	Description
4.1.1	Create Account
4.1.2	Sign In
4.1.3	Create Posting
4.1.4	Update Username
4.1.5	Update Password
4.1.6	View All Postings
4.1.7	View All Postings (Map)
4.1.8	Delete Posting

4.1.1 Create Account

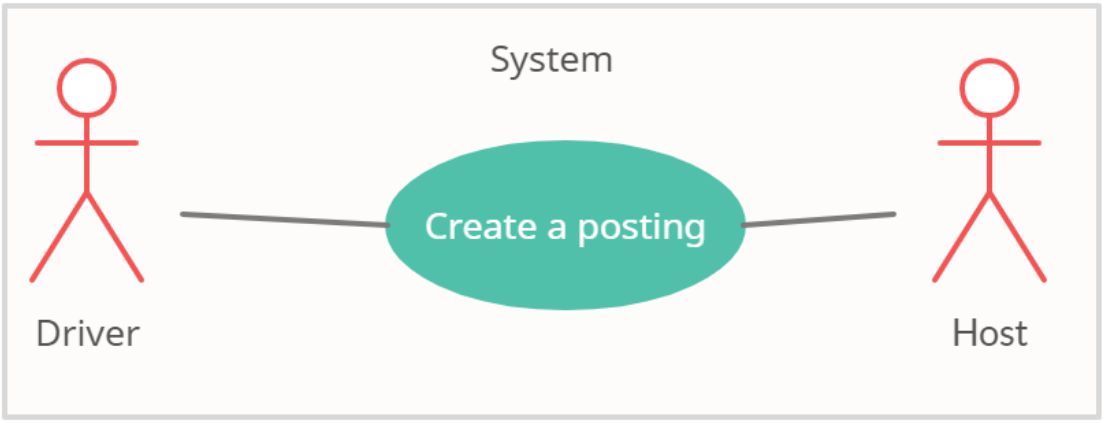


Use Case	Create an account
Trigger	“Create Account” button is pressed by the user upon launch of the app
Precondition	User has the app open and does not have an account created already.
Postcondition	The account is created, the user is returned back to the sign in page and prompted to sign in with the new created account.
Basic Path	Upon launching the app, the user clicks the “Create Account" button which brings them to another page where they are asked to enter their email, password, first name, last name, and phone number.
Exception Paths	If any of the fields are blank, a toast message is displayed, asking not to leave any fields blank.

4.1.2 Log In To Account

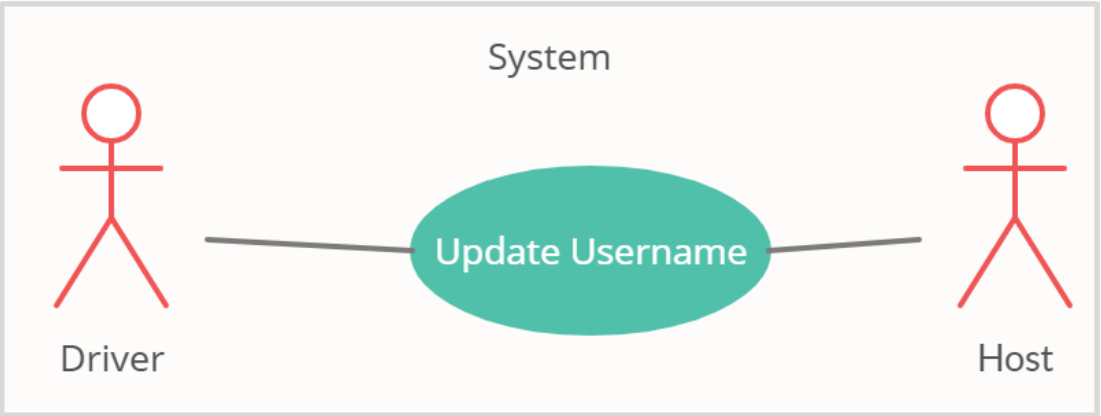
Use Case	Log in to account
Trigger	The user clicks on “Sign In”
Precondition	The user is on the login page of the app
Postcondition	The user is signed into their account, and taken to the homepage of the app
Basic Path	The user launches the app, or logs out of their account.
Exception Paths	If the email and password do not match the data stored in the database, the user must retry their username and/or password

4.1.3 Create posting



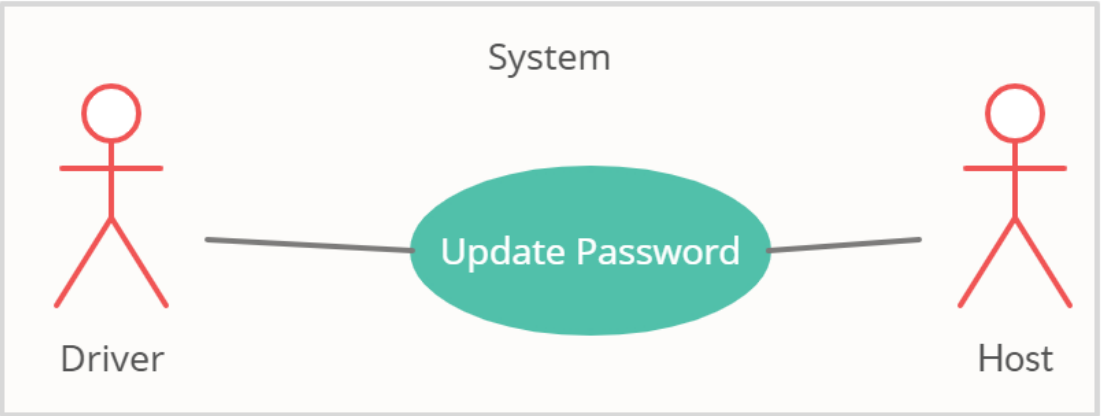
Use Case	Create a posting
Trigger	User clicks on the “Post” button
Precondition	User must be in the Post activity
Postcondition	User is brought back to Main activity
Basic Path	After logging in, from the Main activity the user can select Post, which will bring them to the Post activity to fill in the blank fields
Exceptions Path	If the user does not fill out all of the necessary fields they will not be able to progress

4.1.4 Update Username



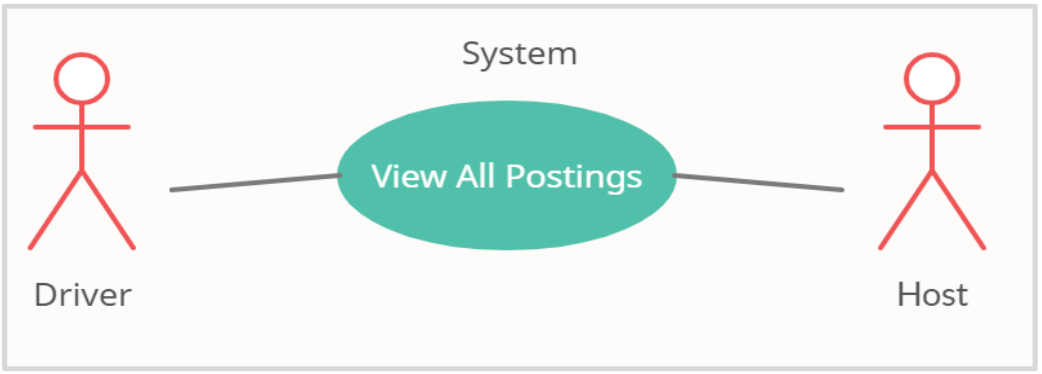
Use Case	Update Username
Trigger	User clicks the “Update Username” button
Precondition	User must be in MyAccount activity
Postcondition	The user is brought back to MyAccount activity, with the “Email” textview displaying the recently changed username
Basic Path	User logins into account, then from the main menu the user clicks on the “MyAccount” button, from there they click on the “Update Username” button
Exceptions Path	If the user does not enter a new username when prompted, they will not be able to proceed back to the MyAccount page

4.1.5 Update Password



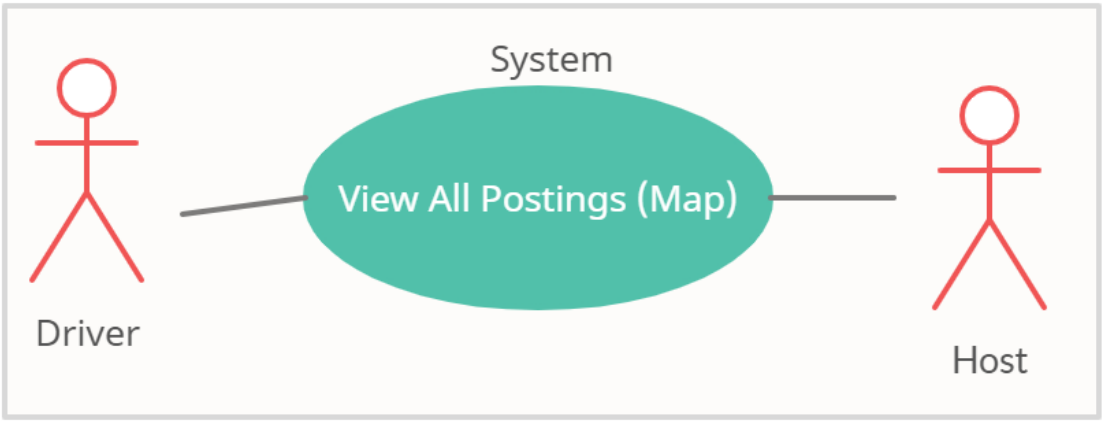
Use Case	Update Password
Trigger	User clicks button labeled “Reset”
Precondition	User must be in the MyAccount activity
Postcondition	The user is brought back to MyAccount activity, with the “Password” textview displaying the recently changed password
Basic Path	User logins into account, then from the main menu the user clicks on the “MyAccount” button, from there they click on the “Reset” button to bring up a dialog box which asks for the current password and new password.
Exceptions Path	If the user typed in the current password, and it does not match the password stored in the database, then a toast message is displayed asking to retype the passwords.

4.1.6 View All Postings



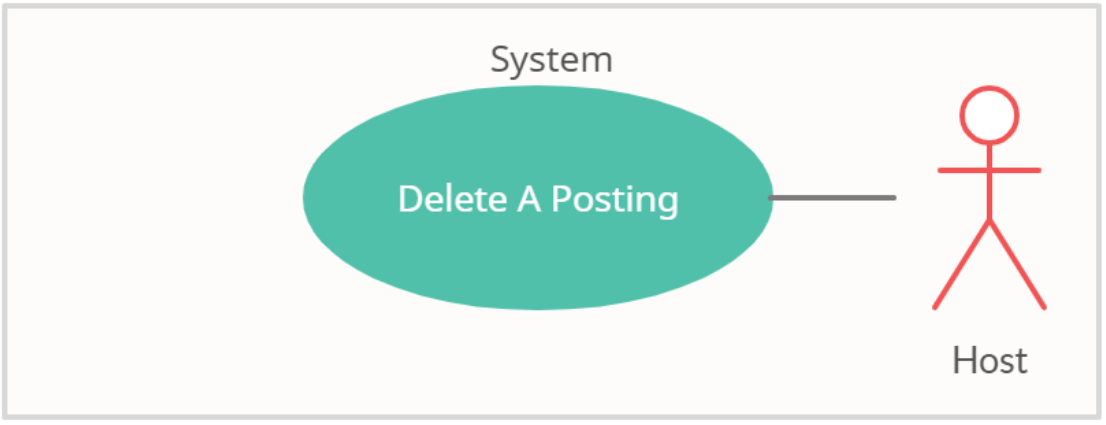
Use Case	View all postings
Trigger	User clicks on the “All Postings” button
Precondition	From the main menu they must select the “All Postings” button
Postcondition	The user will have displayed listview of all posts
Basic Path	After logging in and being brought to the main menu, the user simply has to click “All Postings”
Exceptions Path	None

4.1.7 View All Postings(Map)



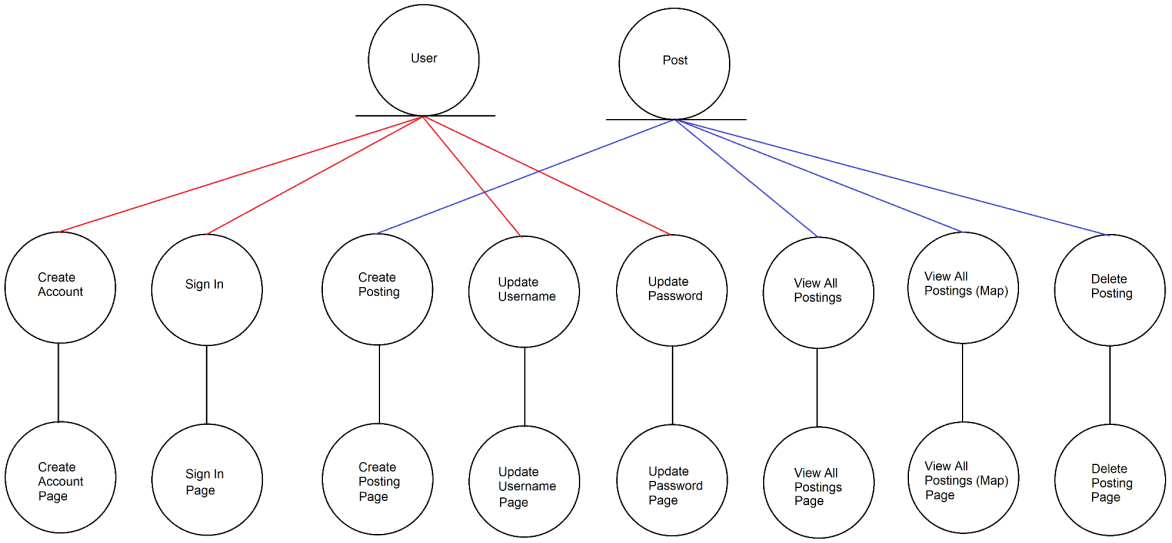
Use Case	View all postings (Map)
Trigger	User clicks on the “All Postings Map View” button
Precondition	From the main menu they must select the “All Postings Map View”
Postcondition	The user will have displayed a google map of all posts represented as either green and/or red pins
Basic Path	After logging in and being brought to the main menu, the user simply has to click “All Postings Map View”
Exceptions Path	None

4.1.8 Delete Posting



Use Case	Delete A Posting
Trigger	User Clicks on the “Delete” button inside of MyPostings after selecting a post
Precondition	From the main menu the user must select the MyPostings, the user must select the posting they wish to delete, then they will be prompted with two options “Delete” and “Done”
Postcondition	The user will be brought back to the main menu
Basic Path	After logging in and being brought to the main menu, the user clicks “All Postings Map View”, from there they select the post
Exceptions Path	If they select the “Done” button instead, it will bring them back to the main menu

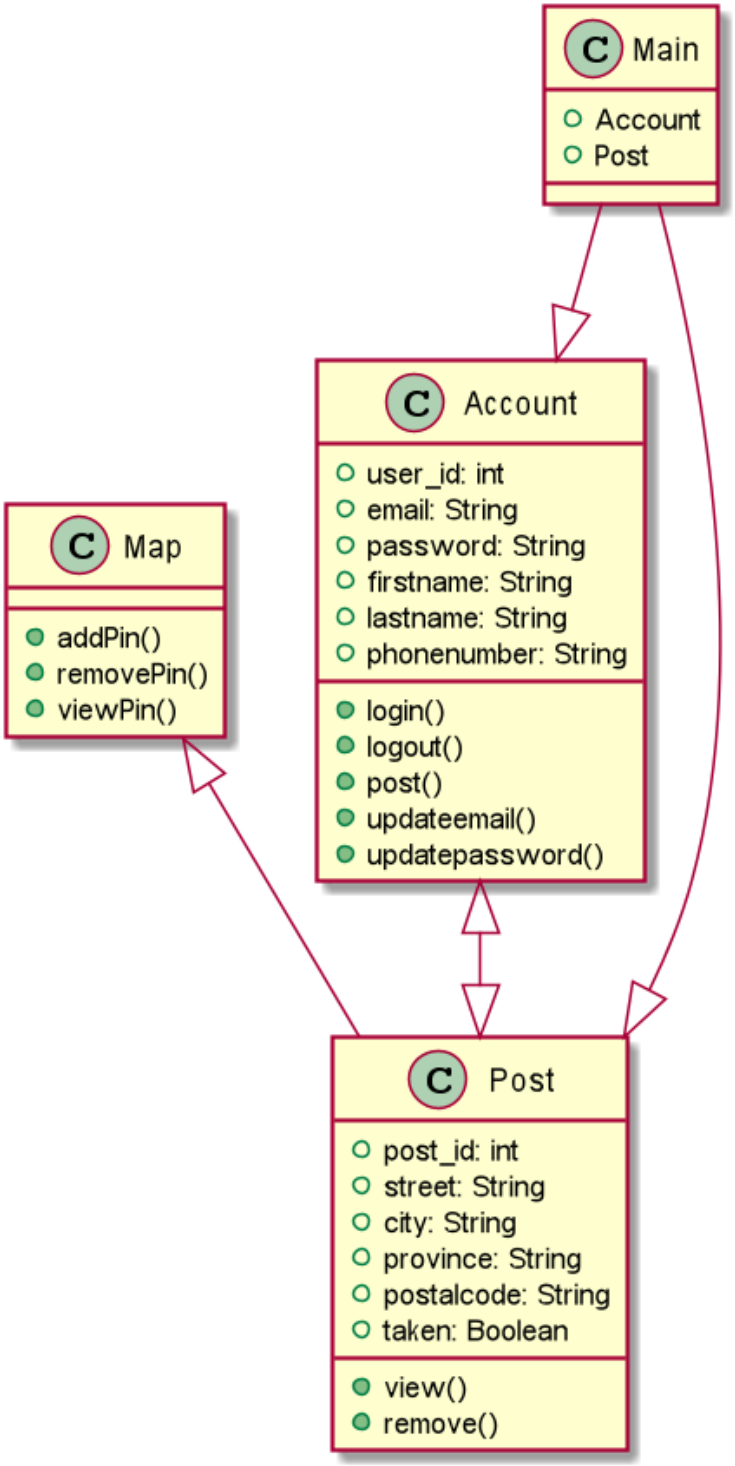
5. Object Diagram



6. Non-Functional Requirements

Requirement No.	Type	Description
6.1	Performance	The application should be efficient and respond in a timely manner compared to other native android applications on the user's device. Regardless of the device the application performance and user experience should be the same.
6.2	Security	Users information is only accessible through the database which is controlled by the developers and only accessible through the app once you login.
6.3	Quality	Posts are shown in chronological order to the user and also shown geographically through google maps API. All posts show correct information as it was entered by the user and more detailed information can be accessed through the list view.

7. Class Diagram



7.1 Main Class

Purpose

The main class is the top-level of the program, which has methods to create Users and Posts..

Prototype: Main()

Called By

- N/A (This is the top-level of the program)

Calls

- Account
- Post

Algorithm

Creates User, and Post instances as needed.

7.2 Account Class

Purpose

The account class holds all user information and has methods to login, logout, create a new posting, and update the account credentials (email and password).

Prototype: Account()

Called By

- Main()
- Post()

Calls

- Post

Algorithm

Creates User account, provides login/logout functionality, assigns new posts made by a user to their account (sets “Host” attribute in database), allows user to update their email address and password.

7.3 Post Class

Purpose

The post class holds all post information and has methods to delete a post and to view a posting in further detail than provided in the listviews on the “all postings” and “my postings” pages.

Prototype: Account()

Called By

- Main()
- Account()

Calls

- Account
- Map

Algorithm

Creates Post, provides functionality to display all details of a post by retrieving the appropriate info from the database (street address, city, province, postal code, taken, and postID). Also allows the user to delete a post they have made and wipes all the post’s data from the database so that it is no longer visible to any user.

7.4 Map Class

Purpose

The map class refers to post information from the database and using the Google Maps API, it adds appropriately coloured pins to the map at the correct locations. Pins can be clicked on to reveal the street address of the post associated with it. When a posting is deleted by the user its record is deleted from the database and thus its pin is removed from the map.

Prototype: Map()

Called By

- Post()

Calls

- N/A

Algorithm

Takes Post records as input and utilizes the Google Maps API which adds, and removes pins and responds to the user clicking on pins and navigating the map by zooming and scrolling.

8. Testing

Date	12/5/2021				
Test ID	Purpose	Steps	Input	Expected Output	Actual Output
Create Account	Testing to see that users can create an account as intended Make sure all sign ups include email and password Make sure sign ups with only username and password are still accepted	Hit the “create account” button Accept the terms of service Fill out the account creation page Hit the “create account” button	None of the field filled out Only email and password filled out All fields filled out	The app will show a toast message saying “Email and Password fields cannot be blank!” The application will allow the account to be created and show a toast message saying “Successfully registered!” The application will allow the account to be created and show a toast message saying	The app will show a toast message saying “Email and Password fields cannot be blank!” The application will allow the account to be created and show a toast message saying “Successfully registered!” The application will allow the account to be created and show a toast message saying

				“Successfully registered!”	“Successfully registered!”
			User enters a taken Email	The application does not allow the account to be created and shows a toast message saying “Account with this username already exists!”	The application does not allow the account to be created and shows a toast message saying “Account with this username already exists!”
Log into account	The user is able to log into an account they created Ensure that If a field in the login is left blank it will tell the user Ensure that If the user logs in with invalid credentials	User enters there login information User clicks the “Login” button	The user hits the login button without enter all fields	The application will not let them login and will show a toast message saying “Please don't leave any fields blank!	The application will not let them login and will show a toast message saying “Please don't leave any fields blank!
			The user enters invalid credentials	The application will not let the user login and show a snackbar message saying “incorrect	The application will not let the user login and show a snackbar message saying “incorrect

	it will tell the user		The user enters the correct credentials	password or username!“ The account will let the user login and show a toast message saying “Login Successful!”	password or username!“ The account will let the user login and show a toast message saying “Login Successful!”
Using the main menu	Ensure that each button in the main menu sends you to the correct	User clicks the different buttons in the main menu	The user clicks the “post” button The user clicks the “all postings” button The user clicks the “all postings map view” button The user clicks the “myaccount” button The user click the “myposting s” button The user clicks the	The user is sent to the create a post page The user is sent the the all postings page The user is sent the the all postings map view page The user is sent the the my account page The user is sent to the my postings page The user is logged out of	The user is sent to the create a post page The user is sent the the all postings page The user is sent the the all postings map view page The user is sent the the my account page The user is sent to the my postings page The user is logged out of

			“logout” button	there account and sent to the login page	there account and sent to the login page
Making a post	The user is able to create a post as intended Ensure that the information from the edit texts are being inserted into the SQL database properly Ensure that the user is not able to post if they do not enter all the information	User enters the in the edit texts for making a post User selects the availability in the check box User clicks the “post” button	The user does not complete the form needed to make a post --- The user enters all the information needed for a post	The application will not let them post and will stay on the post creation screen --- The application will let the user post and send them back to the menu page	The application will not let them post and will stay on the post creation screen --- The application will let the user post and send them back to the menu page
Map View Validation	The user is able to view the posts that they and other users have made on the map	User clicks the “All Postings Map View” button Once in the map the user views the	The user opens the map and clicks on a pin	The user clicks on a pin and then above the pin it displays the street address at which the pin is located	The user clicks on a pin and then above the pin it displays the street address at which the pin is located

	Ensure that the color of the pin in the map correspond s to the availability Ensure that the address information entered by the user shows up when selecting a pin	different pins The user clicks on the pin and it shows the correspon ding address for that pin			
List View Validation	The user can view their posts and active posts in the two listviews When selecting a post a custom dialog shows up giving detailed information about the post Deletion of posts deletes	User clicks on the “My Postings” or “All Postings” button Clicks element in list Custom dialog pops up If in “My Postings” then the user clicks the delete button	The user selects a post from the “My posts” listview and when the custom dialog pops up, they click “delete”	When in the listview the user selects an item and the custom dialog pops up with detailed information about the post and 2 buttons “delete” and “ok” if they select “delete” they will be sent back to the main page and the item will be deleted	When in the listview the user selects an item and the custom dialog pops up with detailed information about the post and 2 buttons “delete” and “ok” if they select “delete” they will be sent back to the main page and the item will be deleted

	them from the list view, map and the database correctly		The user selects a post from the listview and when the custom dialog pops up they click outside of the bounds of the dialog	When in the listview the user selects an item the custom dialog will pop up showing detailed information about that post and when they click outside of the dialog, it disappears and they return to viewing the list of posts	When in the listview the user selects an item the custom dialog will pop up showing detailed information about that post and when they click outside the dialog, it disappears and they return to the listview of posts
Account Information updating	Ensure that the user is able to update their account information Make sure that the updated information shows up in the my account page Make sure that the updated information	User clicks the “My Account” button User clicks either the “reset username” or the ”reset” button User enters their updated information	User presses the “reset username” button and enters their old email as the new email --- User presses the “reset” button and enters their old password as the new password	The email is updated to itself, thus not changing. A toast message displays “Email successfully updated!” --- The password is updated to itself, thus not changing. A toast message displays “Password has been updated”	The email is updated to itself, thus not changing. A toast message displays “Email successfully updated!” --- The password is updated to itself, thus not changing. A toast message displays “Password has been updated”

	is added to the database		User presses the “reset username” button and enters a blank string as the new email	The username is not updated and the dialog remains on the screen so that they can retry.	The username is not updated and the dialog remains on the screen so that they can retry.
			User presses the “reset” button and enters the wrong password as the old password	The password is not updated and a toast message displays “Current password entered does not match, please try again”	The password is not updated and a toast message displays “Current password entered does not match, please try again”
			User presses the “reset” button and enters the correct password as the old password and a valid password as the new password	The password is updated and a toast message displays “password has been updated”	The password is updated and a toast message displays “password has been updated

			User presses the “reset username” button and enters a valid email as the new email	The email is updated and a toast message displays “email has been updated”	The email is updated and a toast message displays “email has been updated”