## Cmput 291 Fall 2018 - Mini Project 1

(a) System Overview

   This is a system that handles data in a databse for a ride sharing service. In this application, members are able to register an account or log in to an existing one, using a unique email address. The app allows members to make an open ride offer that other members can see, in which they can message the offering membering in order to book seats on the ride. That being said, members are able to search for rides using specific keywords, with the app displaying all the search results. Members are able to send and receive messages from other members, and respond accordingly, for example booking seats on a ride they offer. As well, members are able to post ride requests in the event that they do not find rides that match their travel plans. In this case, members are able to search up ride requests, and in turn message the requester about a new ride offer matching the request. Moreover, members are able to manage their bookings and ride requests, with the option to delete any booking or request.

   On each interface, there are prompts that hints user on what they are able to input, and informs them if their input is invalid. This straightforward way expects users to be less confused with the functionality that the app provides, allowing smoother flow between activities.

   In general, the app flows as follows:
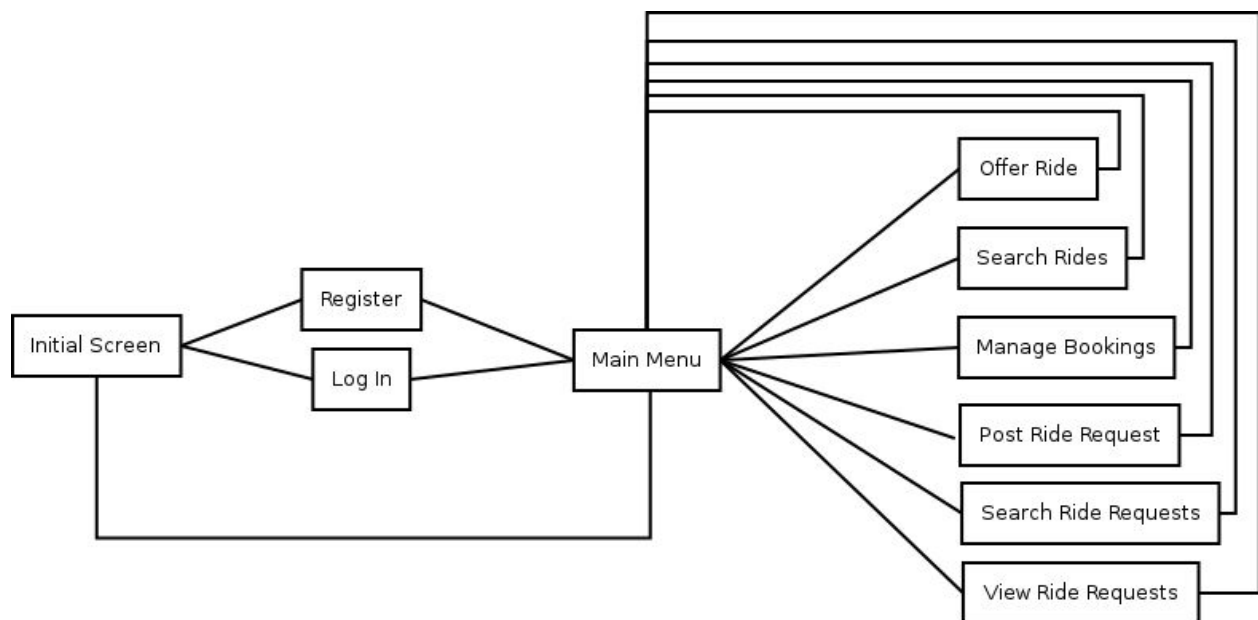
(b) **Activity Functionalities**

detailed design of your software with a focus on the components required to deliver the main functions of your application

(Initial screen) - This is the opening interface upon first boot. In this activity, user is prompted to either Log in or Register.

(Log In) - User is prompted for an e-mail and a password, which the app then checks if it exists in the databse with the right password for the account. On 5 failed tries, the app brings the user back to Initial Screen.

(Register) - User is prompted for information in order to create a new member. First, the email provided is checked, and in the case that the email already exists in the database, user is prompted for another. After all information is collected, new member is saved and is brought into Main Menu.

(Main Menu) - Initially, member is shown the inbox, displaying any new unread messages or stating no new messages to be displayed. The main functionality of this activity is providing user access to other functionalities as follows.

(Offer Ride) - User is prompted for all information required for making a new ride offer, as well as asking for an optional car associated and enroute locations. After, a new ride offer is created with the user as the driver. The user is then redirected to Main Menu.

(Search Rides) - The user is prompted for keywords regarding a location search. The user is prompted for all information regarding booking an existing ride offer. This activity handles inputs and errors and allows members to message drivers regarding a ride offer.

(Manage Bookings) - This activity allows user to view his own bookings and cancel any. Also, it allows user to view any request that applies to user's ride offers and select members to book.

(Post Ride Request) - This activity prompts user for all information regarding posting a new ride request, and handles saving new request onto database.

(Search Ride Requests) - This activity prompts for keywords and displays requests that coincides with the search. This also handles messaging a requester regarding a ride offer by the current user.

(View Ride Requests) - This activity displays all ride requests the user has made, and gives the user the option to delete any.

All activity is connected by the Main Menu, meaning any activity can be reached through the Main Menu. The app opens with a prompt asking for the option to either Log In with an existing account or Register a new account. Else, each activity handles their own responsibilities, only connecting to the Main Menu.

(c) **Testing strategy**

Using python, our program is split into separate larger functions on each functionality requirement, with a few smaller helper functions that assist with the program overall. We are to test each function individually, and assume that other functions are not affected by any and all problems in another, unless said function is called by the initial function, i.e. OfferRide() is not affected by bugs in SearchRides() while MainMenu() would be. With this, each member working on their respective functions will not affect another member's work.

(d) **Group work break-down strategy**

The following are the project functionalities and the group member(s) who worked on it.
- (1) UI design   -harrold
- (2) Initial SQL connection   -bo5
- (3) Log In   -bo5, harrold
- (4) Register   -bo5
- (5) Main Menu   -harrold
- (6) Inbox   -harrold

(7) OfferRide   -bo5

(8) SearchRides   -sun6

(9) ManageBookings   -sun6

(10) PostRideReq   -bo5

(11) SearchRideReq   -harrold

(12) ViewRideReq   -harrold

(13) Helper functions   -bo5, harrold

On average, each member has spent the following hours on this project:

bo5   - 10 hrs

harrold   - 20 hrs

sun6   - 10hrs