

ECE1778

RIDELET

CARPOOL APPLICATION

ECE1778

Syed Hassan, 991680579

Rija Javed, 994575410

Sharon Lourduraj, 993833107

Report Word Count: 2203

Apper Context Word Count: 299

1. INTRODUCTION

DESCRIPTION

This is the final report of RideLet, an Android carpooling management application. Users can take on the role of a driver or a passenger depending on the carpool, wherein the driver of a car creates a carpool posting with details such as time, location, available seats, etc and passengers can sign up for these postings based on the details posted.

MOTIVATION

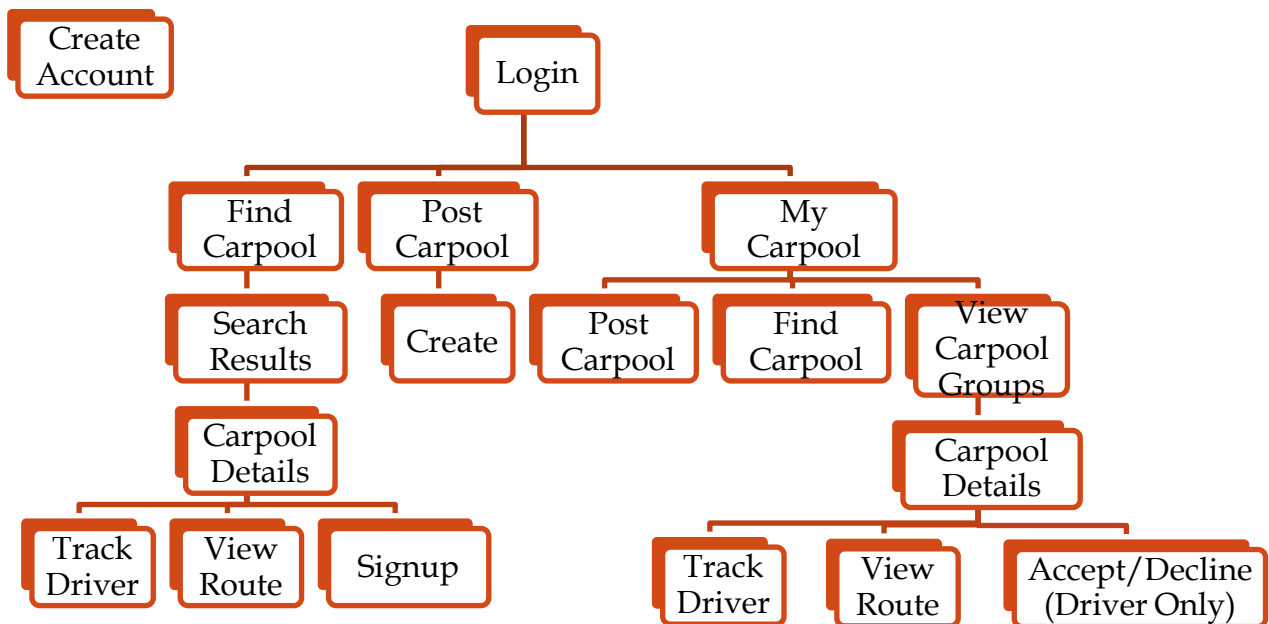
This application is a great tool for those who commute on a regular basis and wish to decrease their transportation costs as well as opt for a more environment friendly solution. Lastly the application is also useful for anyone who needs a ride to get to a particular location, be it once or a recurring event.

2. OVERALL DESIGN

CLIENT

BLOCK DIAGRAM

The following diagram outlines the basic user workflow in the application.



Upon entry the user can either login or create a new account. Once a registered user logs in, they can search for carpools, create a new carpool posting or look at their carpools.

MY CARPOOL

This page allows the user to look at all the carpools that they have signed up for, create a new carpool posting, or find a carpool. The signed up carpools are the ones in which the user is the driver, or a passenger whose status is either accepted or pending (i.e. the driver has not accepted or declined them). Each carpool can be viewed in more details on their page wherein a driver can accept or decline passengers, a passenger can sign up, and any user can view the route on a map or as directions. The details page also contains other specifics like time, date, and to/from location. Finally from this page a driver can allow themselves to be tracked and a passenger can view the driver's physical location on the map.

FIND CARPOOL

The user fills out the required fields, and our application then returns the matching carpools for the query. The user can view those carpools in more detail on their specific pages. Finally, if they wish to do so they can also sign up for that carpool.

POST CARPOOL

A user taking on the role of a driver can create a new carpool posting by specifying details like the to and from location, time, date, available passenger seats, etc.

XML MESSAGE BUILDER/PARSER

The communication between the application and the server involves passing information in XML format. This component provides the ability to build XML messages from native application objects or translate XML responses from the server to the application native objects.

CARPOOL WEB SERVICE CLIENT

Carpool web service client is the gateway to the central server to exchange information regarding carpool postings. It communicates to the server over HTTP and leverages XML format for passing information.

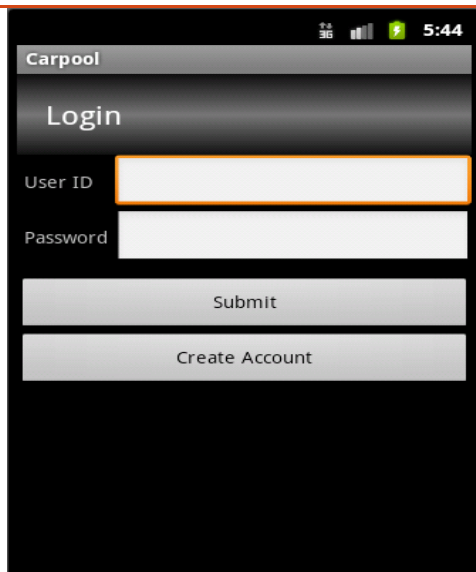
SERVER

DATABASE

We store the necessary data in a relational database. Given the simplicity of the application and the database structure, scalability is not an issue. The database stores interpreted and finalized data as it is submitted to the server, so that when the client requests data, it can be served quickly without any further processing.

3. STATEMENT OF FUNCTIONALITY

Our application allows users to create postings for carpools as drivers and to search and sign up for these carpools as passengers, granted the driver accepts them. Each carpool has a detailed page describing the specifics such as time, location, recurring occurrence, etc and an option to view the route on a map and as directions. Lastly drivers can allow themselves to be physically tracked by the passengers. Below we describe each screen in more detail.



LOGIN

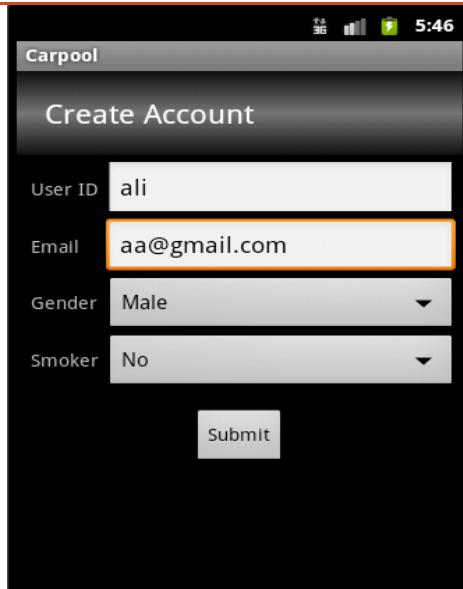
User enters username and password to login into the application.

FUNCTIONALITY

- The application retrieves the user information entered and stores it in Shared Preferences.
- Create Account button forwards the user to Create Account screen.

CONSTRAINTS

The actual authentication based on password is not implemented

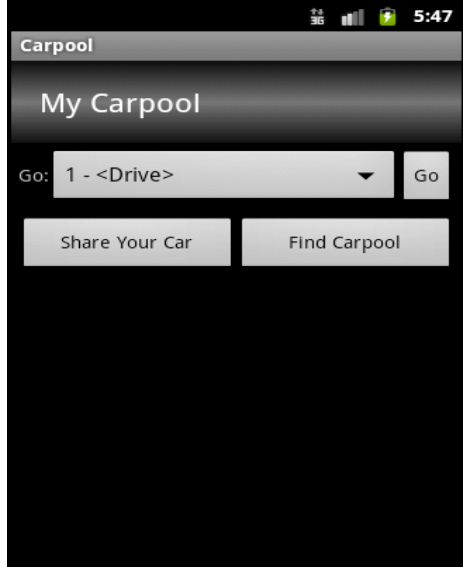


CREATE ACCOUNT

Allows the user to create a new account.

FUNCTIONALITY

When submitted the application publishes this information to the server and creates an account.



MY CARPOOL

Home screen for the application.

FUNCTIONALITY

- The application populates the dropdown with all the carpool postings and pickup requests submitted by the user by querying the server.
- Go button forwards the user to details screen for carpool posting selected.
- Share Your Car button forwards to creating a new carpool posting screen.
- Find Carpool button forwards to the finding a carpool screen.

Carpool

Share My Car

From

To

Departure Time

Passengers ▼

Sign-up deadline

Luggage ▼

Recurring ▼

Notes

SHARE MY CAR

Allows the user to create a new posting by entering detailed information about the carpool.

FUNCTIONALITY

- Once submitted the application publishes this information to the server.
- Selecting Recurring to "Yes" brings up the Day Picker Dialog that allows the user to choose the recurring days.

CONSTRAINTS

The recurring functionality is not fully implemented due to time constraints.

Pick days

Sun	<input type="checkbox"/>
Mon	<input checked="" type="checkbox"/>
Tue	<input checked="" type="checkbox"/>
Wed	<input type="checkbox"/>
Thu	<input type="checkbox"/>

Done

DAY PICKER DIALOG

Allows the user to select the recurring days

Carpool

Find Carpool

From

To

Departure Time

Sign-up deadline

> Advanced Options

Passengers ▼

Gender ▼

Smoker ▼

Luggage ▼

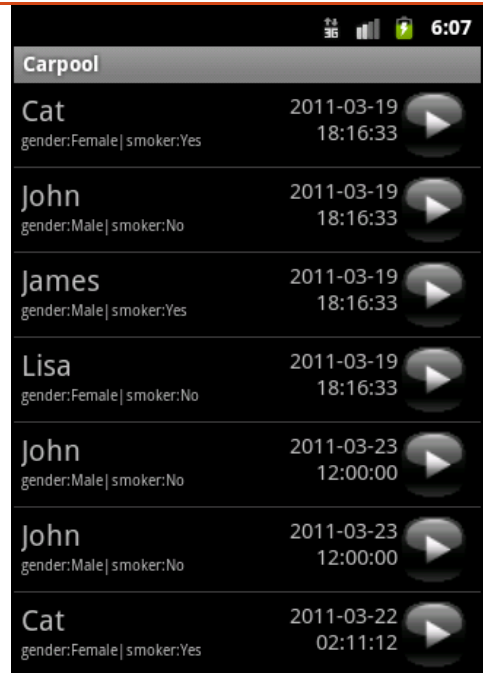
Recurring ▼

FIND CARPOOL

Allows the user to find a carpool posting based on the information entered.

FUNCTIONALITY

- The user can enter more details by clicking on the Advanced Options tab.
- Once submitted the application forwards the user to View Carpool Postings screen



VIEW CARPOOL POSTINGS

Shows the list of carpool postings to the user

FUNCTIONALITY

- Based upon the user selected criteria on Find Carpool screen, the application queries the server for carpool postings.
- The retrieved results are displayed as a list showing the driver information in addition to the departure time.
- The black button forwards the user to the View Carpool Details screen.

CONSTRAINTS

The search algorithm is not fully implemented so all of the carpool postings are retrieved

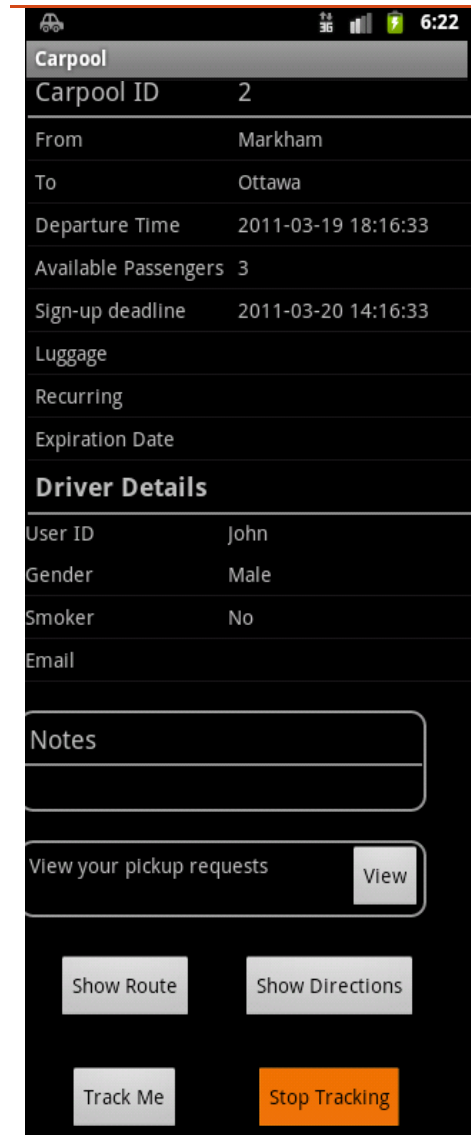
Carpool ID	2
From	Markham
To	Ottawa
Departure Time	2011-03-19 18:16:33
Available Passengers	3
Sign-up deadline	2011-03-20 14:16:33
Luggage	
Recurring	
Expiration Date	
Driver Details	
User ID	John
Gender	Male
Smoker	No
Email	
Notes	
View your pickup requests <input type="button" value="View"/>	
<input type="button" value="Show Route"/>	<input type="button" value="Show Directions"/>
<input type="button" value="Sign-Up"/>	<input type="button" value="Show Driver"/>

VIEW CARPOOL DETAILS (PASSENGER)

Shows the passenger the details for the carpool posting.

FUNCTIONALITY

- The definition of the fields is as they are defined in the Share My Car section.
- View button forwards the user to the View Pickup Requests screen.
- Sign-up button forwards the user to the Signup screen.
- Show Driver button forwards the user to the Show Driver screen.
- Show Route/Show Direction buttons forwards to the View Route or View Directions screen.

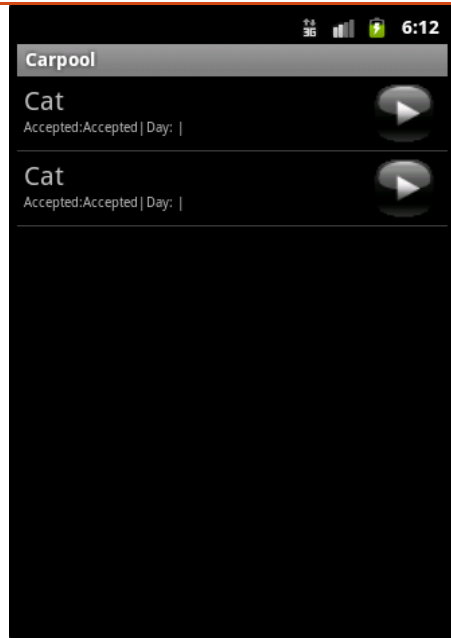


VIEW CARPOOL DETAILS (DRIVER)

Shows the driver details for the carpool posting.

FUNCTIONALITY

- Same as the page for the passenger except there is no Sign Up and Tack Driver button.
- Track Me button starts a Tracker Service in a background that gets the current GPS location of the driver every 30 seconds and publishes it to the server.
- While the GPS Tracker Service is running a notification icon (car) shows up in the top left corner of the screen
- Stop Tracking button stops the GPS Tracking Service



VIEW PICKUP REQUESTS

Shows the list of pickup requests to the driver/passenger

FUNCTIONALITY

- For driver, this screen displays all of the pickup requests
- For passenger, this screen displays the pickup request submitted by the passenger
- Selecting the black button forwards the user to View Pickup Details screen

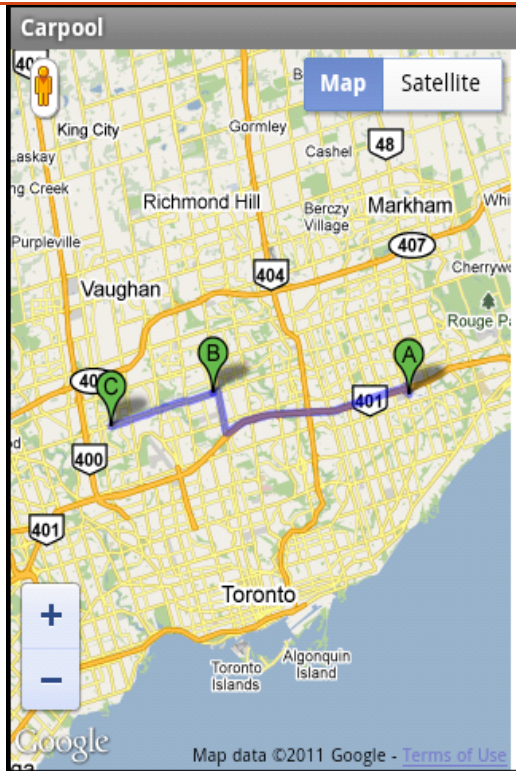


VIEW PICKUP DETAILS (DRIVER)

Shows the details of a pickup request

FUNCTIONALITY

- Carpool ID section lists the specifics of the carpool whereas the Personal Details section lists the information related to the requesting passenger.
- Driver accepts/rejects the pick-up requests using the Accept/Decline buttons
- Accept and Decline button are visible only when the accepted pick-up requests are smaller than carpool capacity.



VIEW ROUTE

Shows the route for a specific carpool on a map.

FUNCTIONALITY

- The start and end destination on the route are the driver's to and from location
- The stops on the route (such as B on the map) describe the accepted passenger(s)' destinations.

Carpool

A 300 Consilium Pl, Toronto, ON M1H, Canada

17.1 km - about 16 mins

1. Head north on Consilium Pl toward Corporate Dr	0.2 km
2. Take the ramp to ON-401 W	0.4 km
3. Merge onto McCowan Rd	0.2 km
4. Take the ramp onto ON-401 W	12.8 km
5. Take exit 369 for Yonge Street N	0.5 km
6. Merge onto Yonge St	2.7 km
7. Turn left at Finch Ave W Destination will be on the right	0.4 km

B 58 Finch Ave W, Toronto, ON M2N 2H1, Canada

7.4 km - about 11 mins

VIEW DIRECTIONS

Shows the directions for a specific carpool.

FUNCTIONALITY

- The start and end points as well as the stops are as described for the View Route screen.

Carpool

Pickup Details

Carpool ID 25

From jhbv

To hhff

Departure Time

Day

Status Pending

Luggage

Personal Details

User ID q

Gender Male

Smoker No

Email

VIEW PICKUP DETAILS (PASSENGER)

Shows the details of a pickup request

FUNCTIONALITY

- Same as the View Pickup Details (Driver) page except there is no Accept and Decline button.

Carpool

Sign-up

From Yorkdale, Toronto

To West Mall, Toronto

Departure Time 2011-04-02 18:14:58

Select

Recurring No

Luggage No

Sign-Up

SIGN-UP

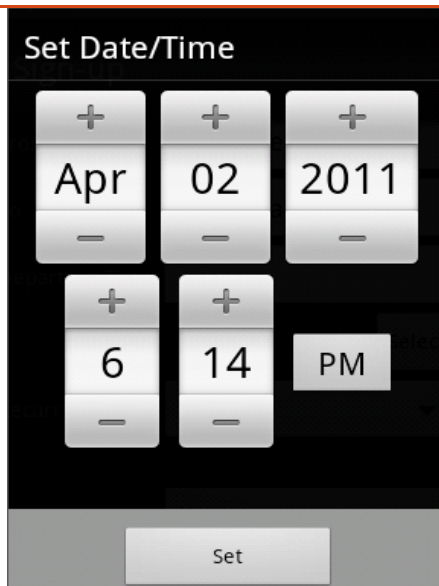
Allows the user to sign-up for a carpool posting by entering the required fields.

FUNCTIONALITY

- Select buttons pops up Date/Time picker dialog box to select a date/time
- Application publishes this information to the server once the user clicks Sign-up

CONSTRAINTS

Recurring pick-up request is not fully supported.

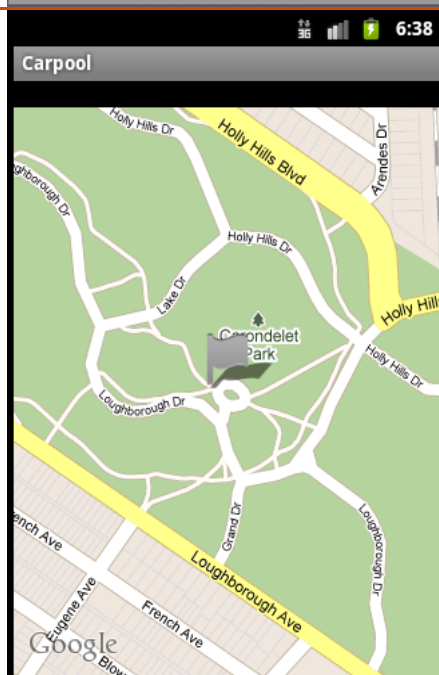


DATE/TIME PICKER DIALOG

Allows the user to select a date & time

FUNCTIONALITY

- A custom widget that allows the user to select the date & time
- All the date fields in the application uses this dialog box to allow the user to pick date/time



SHOW DRIVER

Shows the current location of the driver

FUNCTIONALITY

- Application queries the location of the driver for the selected carpool and displays it on a map.
- Application uses a background thread to query the server every 5 seconds.



MAIN MENU

Application's main menu

FUNCTIONALITY

- Home option navigates the user to the My Carpool page.
- Logout option logs out the current user and shuts down the GPS Tracker Service if it is running

GPS TRACKER SERVICE

Publish the driver's GPS location to the server

FUNCTIONALITY

- Service runs in the background to get the GPS location every 30 seconds and then publish it on the server

CONSTRAINTS

Our web server infrastructure starts rejecting our publish requests due to strict rules against Denial of Service attack. Unfortunately, we do not own the infrastructure therefore we have to adhere to the server provider's policies.

4. LESSONS LEARNED

INITIAL SCOPE

We had under estimated our initial scope because of which we had to remove various features from the scope. The reasons for this discrepancy were that firstly we were not able to factor in the timelines for buying a web hosting site from a web hosting provider and secondly, we underestimated the time required for integration testing. Hence, if we were to do this project again we would factor in these time constrains in our project plan.

WEB SERVER LIMITATIONS

We discovered that policies for the Denial of Service attack are very strict. Therefore, when the application was sending the same request on a periodic basis the server would start rejecting the connection after the first few requests. Thus, in the future we would do more research into the infrastructure policies as well as look Google's App Engine as an alternative.

5. CONTRIBUTIONS

SYED HASSAN

Main contribution was laying down the application's foundational framework which includes a central controller to control the navigation, a web service client that communicates with the server and custom Widgets/Dialogs. In addition, he also developed various features such as tracking the driver's current location, creating/searching a new carpool posting and view/creating new pickup requests.

RIJA JAVED

Major contributions include implementing a routing algorithm to find the best possible route given the physical to and fro addresses for the driver and the passenger(s) as well as the time restrictions and for a given carpool. Along with finding a route, she also added a feature to show directions and a map for this route for accepted passengers, driver and any registered user viewing the carpool details. Lastly she was also involved with integration for the final design.

SHARON LOURDURAJ

Main contributions include implementing the architecture on the project's server components. These components include the database architecture, web server setup and the web service framework; along with getting the physical server up and running. A HTTP based REST protocol was implemented for the web service using PHP and the Zend Framework, with MySQL as the database backend. He also developed an URL abstraction layer that allowed the client to access the server without worrying about the details of how the server handles the data submitted, or requested.

6. APPER CONTEXT BY SHARON LOURDURAJ

Given the unique situation of having a broad background in both System's Design Engineering and Mechanical Engineering, I chose to approach the project from an architectural standpoint using techniques learned in Systems Design Engineering. Looking at the project from a broader perspective, the intention was to build an app that was structured and extensible in nature. This involved ensuring that we performed a bottom up process in the development of the project. First implementing the foundational framework for the client and the server, then the functional features of the

client and the server, and finally bridging the gap between the client and the server by integrating them together. The project achieved the integration of a solution to a real world problem onto the mobile device, using techniques and principles learned in designing systems as a whole. It is influential in the sense that there were more aspects to the project than what could be seen on the surface, which is just software with client-server architecture. It is the design process that is important, the iterative style of bottom-up development allowed us to build on basic features to construct the system. Development of the client software for the mobile phone, and the flexibility of implementing a web service to talk to any client device (mobile phone, or a web interface) demonstrate the concepts of componentization and modularity. If we were to port the application to the iPhone, only the user interface portion of it would need to be rewritten, the business logic and the data model are completely portable. This is what we aimed for during the development of the project, and we have achieved it. Other than that, the project does not add anything new to the Systems Design field, as the project itself is based on its utility.

7. FUTURE WORK

FEATURES

Possible features that can be included to augment our application are:

- **Publish/Subscribe Architecture:** Allowing passengers to post ads looking for a driver, and then implementing an infrastructure to match relevant driver ads with passenger postings. Hence the drivers would be publishers and the passengers would be the subscribers. Nevertheless in this scenario, the publishers and subscribers can interchange their roles based on the content.
- **Security:** Authenticating the users based on their email address, physical address, date of birth, credit card, etc can make our application secure for the users against crime and infringement of privacy.
- **Payment System:** Implementing our own payment system requiring a valid credit card as well as allowing users to conduct compensation exchanges on their own.
- **User Profile:** An interface wherein users can send each other messages, organize their carpools (according to their roles, time and status) and rate one another.

RELEVANCE & COMMERCIALIZATION

This application is relevant for commuters as it lowers their transportation costs, it helps people find a ride, and it is energy efficient as it decreases car pollution by encouraging commuters to ride together in one vehicle. Nevertheless this application can be useful in transportation means other than a car as it can be extended to transport businesses like the TTC, Go Transit, and taxi companies to allow the user more transportation options as well as possibly give the businesses more customers. Extending our application to transportation business would allow commercialization, as would third-party advertisements. More importantly though, the features listed above, especially security, will help commercialize our application.