

# Documentation for Bus Plan

*Created By:*

*Eugene Lee, Dylan Rebelak, Manan Saraf, Rohan Joseph, Steven Sekowski, and David Toh*

---

## Table of Contents

<b>Section</b>	<b>Title</b>	<b>Pages</b>
1	Description	1
2	Process	2-3
3	Requirements & Specifications	3-6
4	Architecture & Design	6-10
5	Future Plans	11-12

## **1. Description**

The Bus Plan Android application uses the CUMTD bus service API in order to provide the people of Champaign-Urbana with accurate and convenient bus service information. The main features offered include a way to search of all the bus stops, a way to favorite stops for quick access, a way to view all nearby stops based on GPS location, a way to plan a trip from a certain stop to another stop, and a unique way to set up scheduled trip reminders. There are many apps out there that offer bus service information, but Bus Plan was designed in order to make the most useful actions easy to perform which can be seen for example in how the favorites are right on the front screen.

## **2. Process**

Our team decided to adopt the same process used in the Software Engineering course taught last semester where the teams followed a programming practice similar to extreme programming, or XP for short. It involves splitting up the software program into smaller features, or use cases, that programmers can focus on and develop independently of other pieces. This allows for the developers to really break down all of the components and see the work that needs to get done and what it will take to complete the project before any work even begins. Another advantage this process offers is that during planning someone can realize dependencies among features and then they can put these features into the schedule accordingly to avoid any potential setbacks. On top of splitting the features up into manageable pieces, this method segments the development cycle into roughly 2 week periods called iterations. These iterations contain a set amount of work, or units, that the team decides upon and must complete each period. The team then sections out and fits the pieces into the iteration schedule based upon a weight in units the team gives to them according to the difficulty and time to complete. During the iterations the team also meets once together and assess how the current progress is going and if needed adjusts the schedule. On top of the iteration meetings, our team met again once per week to keep each other up to date on how development was going and to plan for the future iterations. The

programming was also done in pairs and each iteration our 3 pairs had 12 hours of work to complete. For testing we implemented unit tests for each of the main activities in our app that we could run periodically to check that past functionality was not affected unintentionally. We used the standard Android framework that works similar to Junit testing. To complement those tests, we used a framework called Espresso to do advanced GUI testing to make sure our GUI was performing the way we expected it to.

### **3. Requirements & Specifications**

When development first started, the major features of Bus Plan could be summed up into six primary use cases. These included Bus Stop Statistics, Search Stops, Favorite Stops, Nearby Stops, Trip Planner, and Scheduler. After drafting the initial idea of the project these cases were created to reflect the core functionality of the application. Starting with UC01, this case handles the basic display of a specific stop and what the user will see. The activity displays all the routes coming to a stop and breaks the stop down into child stops so that the user knows which side of the street that the bus will arrive. The information is all gathered from the CUMTD API and the route name and time until arrival is shown to the user. The Search Stops use case provides the ability to find stops based on a substring search. The user can view a list of all stops based on the substring they search by in the search box. When any of these stops are clicked on it takes the user to the Bus Stop Statistics page for that stop. Shortcuts to specific stops can be set with the Favorite Stops feature. This use case handles if a user wants to bookmark or save a specific stop as a favorite, they can do so and it will be in a preferred list on the main window of the application. Clicking on it in this list will again direct the user to the statistics activity. To favorite a stop, the user must search for it in the search activity, and then click on it. Once on the statistics page if the user clicks on the star in the action bar it will add that stop to favorites, light up the star, and notify the user; to delete the stop on that same activity the user can click the star again and it will be removed from favorites. The Nearby stops feature allows the user to see all of the stops around their current location using the Google maps API and allows them to interact

with the map. Clicking on a stop will redirect them to the bus stop statistics page. The rest of the features can be seen below.

After having developed the main functionality of Bus Plan, other useful features became clear and prompted the addition of a few more use cases. These include the Shortcut Widgets, the Draw Bus Routes, and the I-Stop Buses features which can also be seen below.

#### UC01 Bus Stop Statistics

The user first selects a stop for which he wants to view the incoming buses. The app will then send a request to the CUMTD API, requesting the bus statistic information for that particular stop. The API then returns a list which is then displayed of the incoming buses which are coming within the next 30 minutes. The buses are shown in the order they are due to arrive at the stop. The list shows the bus number, it's name, it's final destination and arrival time. It will also show the CUMTD bus stop ID.

#### UC02 Search Stops

The user first starts typing the name of a bus stop he is searching for. As he types, the app looks up the local app database for corresponding bus stop names. It uses regular expressions to match names and dynamically displays the list as the user types. The list is shown in the alphabetical order. The user then clicks on his desired bus stop and this leads him to the bus stop statistics activity for the stop. The Local Database regularly updates the bus stop names with the CUMTD Database.

#### UC03 Favorite Bus Stops

The user will open up the favorite menu and see any previously favorited stops listed. In this window there is a button called “Add Favorites” which when clicked will take the user to a list of all the stops. The user can search through the stops to find specific ones and click on the star next to the stop name to add it to the favorite list. If already starred and the star is clicked, it will be removed from the list. When a stop on the favorite list is clicked, the user will be taken to the bus stop statistics page for that stop.

#### UC04 Nearby Bus Stops

The app user submits their phone's current location from their phone's GPS data by accessing the Nearby Bus Stops utility. This location (possibly a latitude/longitude pair) is submitted to the Google API, which then gets the nearby intersections/landmarks within a reasonable radius. The Google API then searches the MTD Database for possible bus stop locations. The MTD Database returns all valid stops locations to the Google API, which then plots these stop locations on Google Maps and returns this map to the app user.

#### UC05 Trip Planner

The user will enter a source destination, target destination and time for their trip. The application will use the Google Maps API to convert the given locations to latitude/longitude pairs. The coordinates can then be fed to the CUMTD API to plot a route. The corresponding route (consisting of bus transportation and walking) will then be shown to the user on a map (using Google Maps) with a corresponding list of directions.

#### UC06 Scheduler

The user can input different times that they need to be at a specific bus stop and how far in advance that they want a reminder. The events can be made recurring or one time. When the set reminder time is reached, the app will use the users current location and give them a notification that when clicked will open the app and show them what nearby stop they should go to and what route they should take to get to the desired destination.

#### UC07 Widget Support

The user can add widgets to their phone home screen which will function as shortcuts to a specific stop statistics page. When the widget is clicked on for the first time, it has the user select which stop it should correspond to and a color to mark which stop it is for the user to differentiate the different shortcuts.

#### UC08 Draw Bus Routes

The user is able to click on any of the routes on the bus stop statistics page and a map with the current route drawn on it will appear on the screen. The user can see where the bus travels and the current location of the bus on the route.

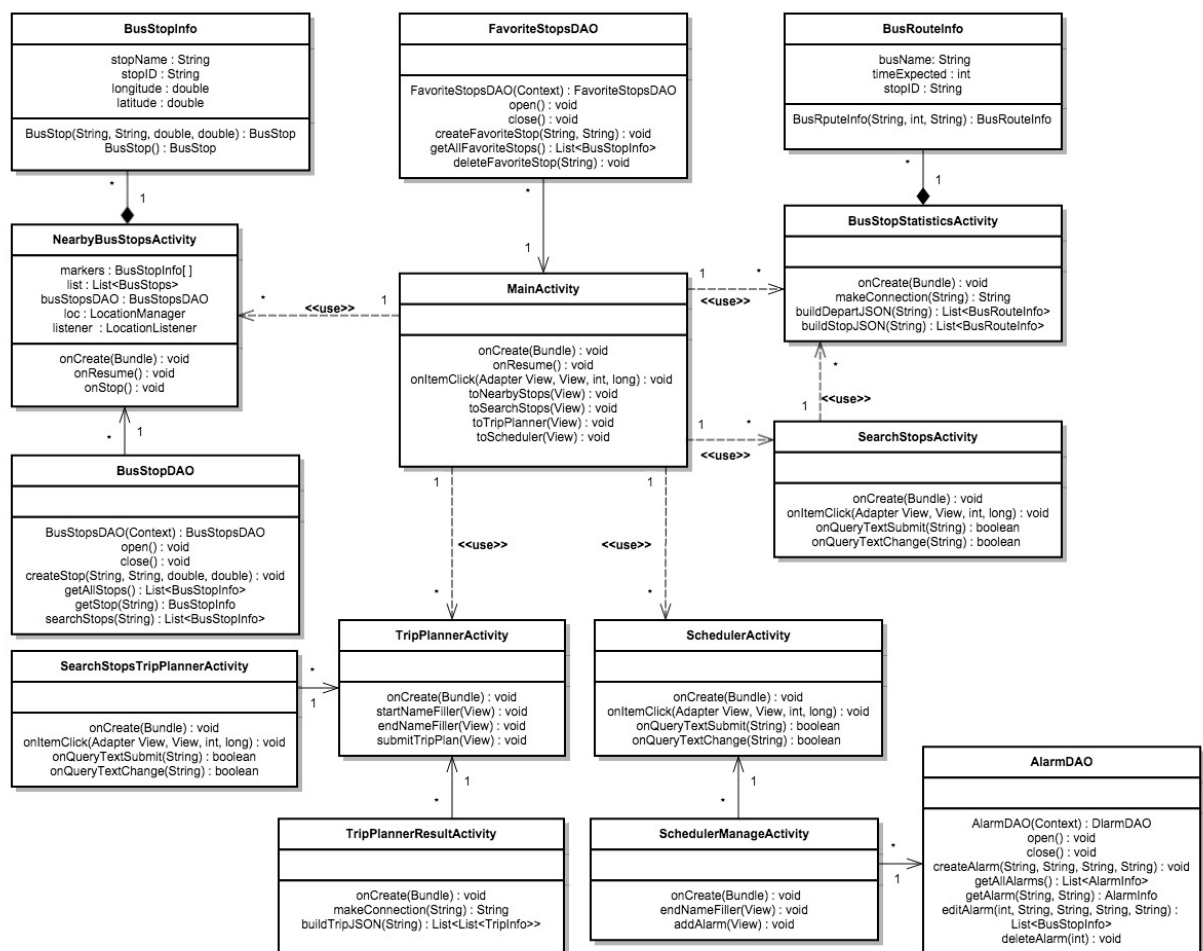
## UC09 I-Stop Buses

When a user opens up the bus stop statistics page, they can see which stops are I-stops based on an icon next to the relevant route names. I-stops usually allow passengers to board bus at that stop without paying any fare.

## 4. Architecture & Design

In terms of functionality, the notable classes in our project can be divided into three basic types: Android activities, database access objects and datatype objects (as well as other minor helper/utility classes). Each android activity generally corresponds to a specific screen of the application

### UML Class Overview



## **Class Overview**

### **-MainActivity**

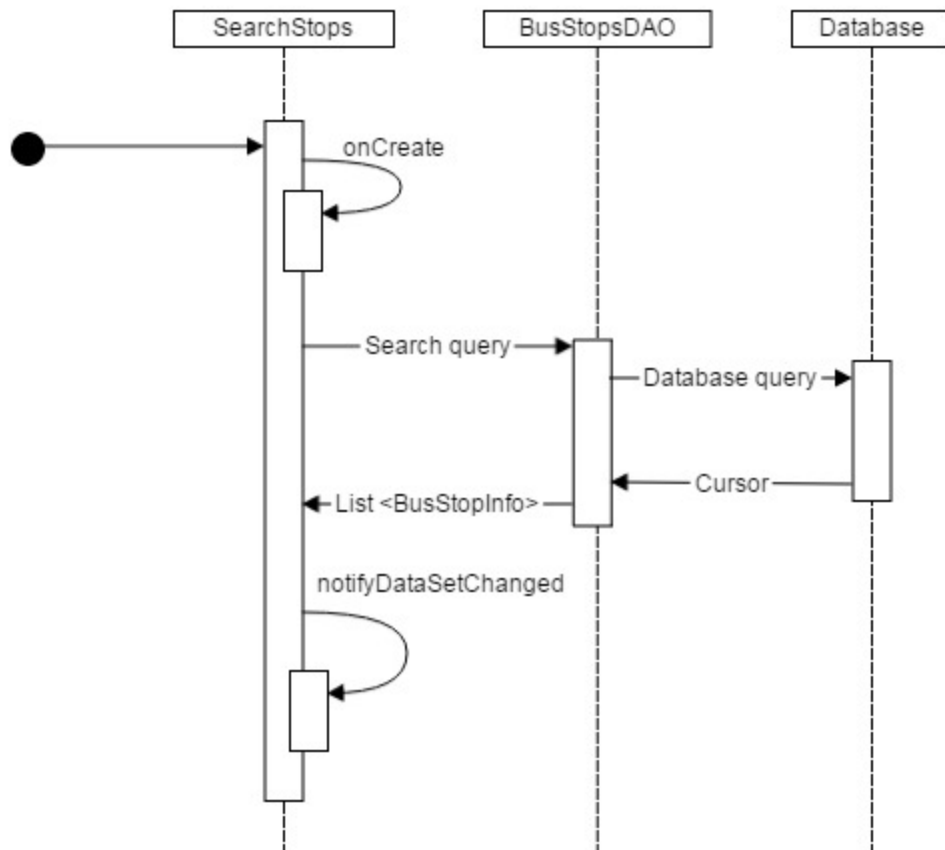
This is the activity that runs when the application is launched and represents the "home screen" of the app. From here the user may click several buttons to transition to other activities/screens (search, nearby stops, scheduler and trip planner) as well as see a clickable list of their favorited bus stops which then opens BusStopsStatisticsActivity for that stop

### **-BusStopStatisticsActivity**

This activity receives input from the activities calling it about the stop name whose information this page should show. This activity uses the cumtd api to receive current information about the buses departing from that particular stop and also returns in how many minutes would each bus arrive. Also the list of buses arriving at the stop is clickable which would lead to DrawRouteActivity which shows the current bus's position and its route.

### **-SearchStopsActivity**

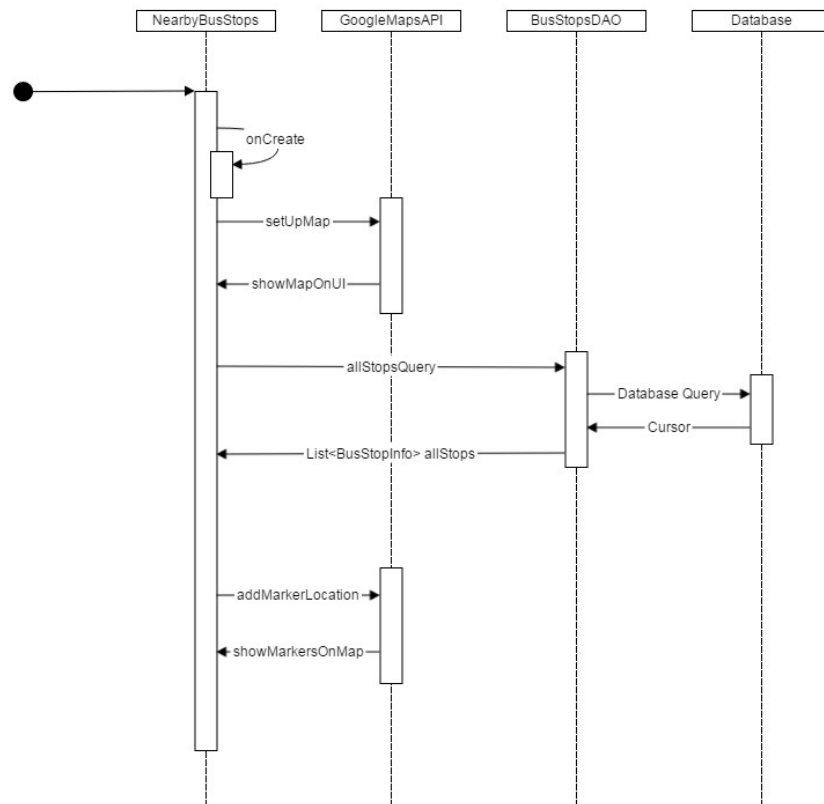
This activity initially displays all bus stops and lets the user filter this list by typing into the search text field located in the actionbar. When the search text field is non-empty the bus stop list will display only those stops whose name contains the search string. Selecting a bus stop in the list opens the BusStopStatisticsActivity for that stop.



### -NearbyStopsActivity

This activity displays a map using Google Maps API centered around the user's current or last known location with all of the surrounding bus stops displayed as markers on the map. The position of the markers is received from the database. If the user clicks a marker the stop name will appear and clicking on the stop name brings the user to the BusStopStatisticsActivity.





### -TripPlannerActivity

This activity allows the user to input a start stop, end stop, and departure time and when they submit this information, it displays the possible trip plans. This information is received from the user and then sent to the CUMTD API which then returns a list of possible trips at that time and then the TripPlannerResultActivity shows these planned trips

### -SchedulerActivity

This activity allows the user to view, edit, and add reminders to the database. These reminders are set up so that 45 minutes before the user supplied arrival time, a notification will be sent to the user and when clicked on will use the current location to plot a trip plan to the user supplied destination.

### **-DrawRouteActivity**

This activity opens up when a user clicks on a specific bus at the BusStopsStatisticsActivity which then shows the specific bus's route by drawing it on a Google Map and then adding a marker to point the current location of the bus they are trying to look for. The map centers itself around the bus's location. We get the route of the bus and the location of the vehicle through the CUMTD API which returns a list of points which we then draw on the map using polygon lines function in the google maps API.

### **Database Overview**

We use SQLite for our database system. We have 4 tables, version, alarms, bus\_stops and favorite\_stops. Table **Version** stores a last feed update date from the CUMTD server. When we launch our application, we compare the date in the database and the date from the server to decide if the **bus\_stops** table needs to be updated. Table **bus\_stops** stores important information of all bus stops. It stores name, id, latitude and longitude of bus stops. We use this table for most of the Use Cases, such as search stops and nearby stops. Table **favorite\_stops** stores information of favorite bus stops of the user. When the user interacts with UC03, this table gets entries added or deleted based on the user action. This table only stores id and name of bus stops because any other data can be gotten from the bus stops table by using the id. Table **alarm** table stores a list of alarms that the user creates. This table stores id, destination, arrive time, days of the week to send the reminder, and repetition of each alarm. In UC06, we send a notification to the user based on what is in the **alarm** table.

## 5. Future Plans

Currently there are no plans to continue developing Bus Plan past this semester. The project was conceived as a convenient way to learn our way around Android and to that respect was a major success. Coming into the project the team did not have any tangible experience with Android, but now as a whole we are fairly comfortable with the language. If our team did decided to release our app on the market, one thing we would probably want to do beforehand would be to distribute our app on a small scale and get some feedback from people outside of our group. As developers, the features and implementations of them seem really solid to us, but we never went outside of our group for advice during the development phase and if we were making this in a real world setting, we should have definitely done that. Overall the project seems successful and operational as a bus service app in its end state.

### Personal Reflections:

**Manan Saraf:** I felt the project taught me a lot of new tools such as Android Studio, using API's and understanding how Android works. The process we used also helped a lot during the development process. Pair programming helped me a lot, first it helped me debug my code a lot faster because another person was always looking over it while I coded it. It also helped us review each other's code and make us realize quicker the mistakes we did while developing the app.

**Rohan Joseph:** This project was one of the most interesting projects that I have worked on. It taught me a great deal as I able to learn something new with Android. I also liked the idea of dealing with multiple API calls and handling many exceptions. The pair programming part of the project was good, as one could schedule meetings easily. I felt that the project was successful as we implemented more than anticipated.

**Steve Sekowski:** I felt that it was refreshing to learn a new style of programming and to be able to integrate my OOP (Java) skills into something practical (an Android app). I felt that the iteration system (for XP) worked well, as for most of the duration of the project the workload was sizeable but not exceedingly demanding. I also felt that pair programming helped in

debugging and solving problems more quickly. Finally, I feel satisfied that we were able to build a feasible Android app that mimics (and in some cases, performs better than) RideCU.

**David Toh:** I learnt a great deal over the course of this project. Starting from zero experience with Android or mobile development meant that most of what we did over the course of the project was new to me. I am happy with the final outcome of the project in terms of the finished application and the skills we acquired during development.

**Eugene Lee :** This project was first time to create an android application for me. I thought this will be tough because I didn't have any experience before. However, I learned lots of stuff from this project and I'm really happy with what I have learned. I didn't follow XP process before this project, but from now on, I really like to follow XP process when I'm with other people to create another program. Because making application was really interesting, I want to learn IOS programming someday.

**Dylan Rebelak:** This project was a fun experience to work with other students and develop a working practical application together. When Eugene and I first proposed the app, we did not really envision it getting carried out, but despite that doubt we got chosen as a project group. Looking at our finished project, it is more than we ever hoped for and I am very happy I got to work with these guys this semester. Through the app I felt that we all got a good understanding of android development and were also able to practice good development strategies.