**Milestone 2 (The detailed UI design)**

**Group Members: Prashant Patel**

**Main Contact’s name and email: ppatel03@syr.edu**

**Find My Bus App**

**Purpose:**

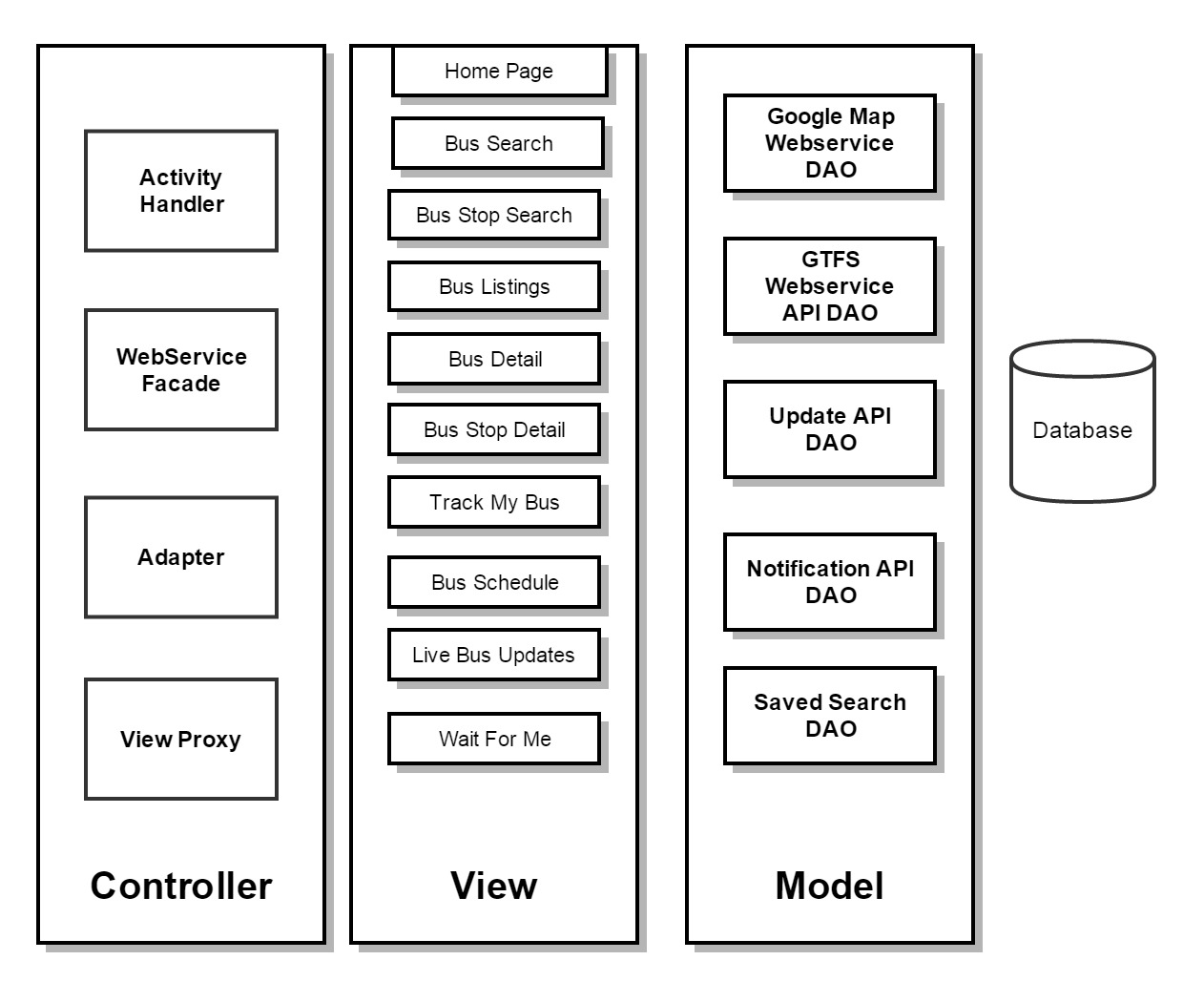
My friends staying in nearby houses face few issues while catching the bus to Syracuse University. Firstly, they need to refer to different time-tables of different buses like Centro, James Street, East Campus, etc. Second, they often waste their time waiting for the bus because it becomes crowded already before reaching to their arrival point.

This App solves the problem by finding all the possible buses and its corresponding status like timings, number of people, delays, etc. reaching to user’s destination after entering their nearest bus stop.

Additionally, App also brings “Wait For Me” Functionality where if an App user is about 30 seconds away from the bus, he can broadcast on the App about his status so that the bus can stop for him (based on bus driver’s courtesy).

**App Architecture:**

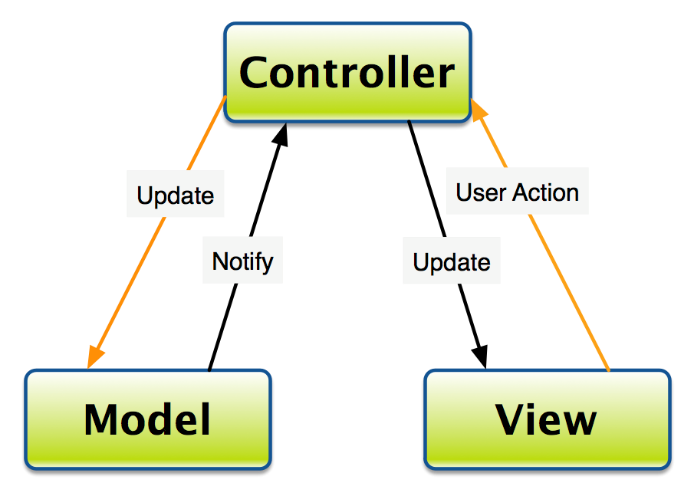
The App follows the MVC design.



MVC Architecture of the App

**MVC Design:**

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.



MVC Design Pattern

* **Model** - Model represents a data access object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
  1. **Google Map Web service DAO**: It is responsible for fetching the required data from the Google Map API, parsing it and returning it as the collection of Java Pojos related to location class.
  2. **GTFS Web Services DAO:**

The General Transit Feed Specification (GTFS) defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way.

This module is responsible for collecting all the required Bus related feeds from GTFS Web services and populate its JSON data into collection of Java pojos related to Bus details class.

* 1. **Update API DAO :**

This module interacts with both GTFS module and the App database to save or retrieve the updates. It requires interaction with the GTFS module because of its real-time updates related to delays, bus tracking etc. However, the App also lays importance to the user’s feedback related to the bus details like delays, current stops, crowded, seats available etc. In order to achieve this, a custom App Level MYSQL database is designed.

* 1. **Notification API DAO :**

The reason why this module is not include in update module is to support highly cohesive functionality of “Wait For Me”. It follows the observer pattern, where on clicking “Wait For Me” calls this module which first notifies the other observers (which are nothing but bus drivers) and accordingly saves the details in the database.

* 1. **Saved Search DAO :**

This module only interacts with the custom database to save or retrieve the search criteria of the user according to different dates. It also requires an implementation of high performance cache based on Service Locator Pattern to provide the cached based recent searches to the users who wanted to get the real-time bus schedules quickly.

* **View** - View represents the visualization of the data that model contains.

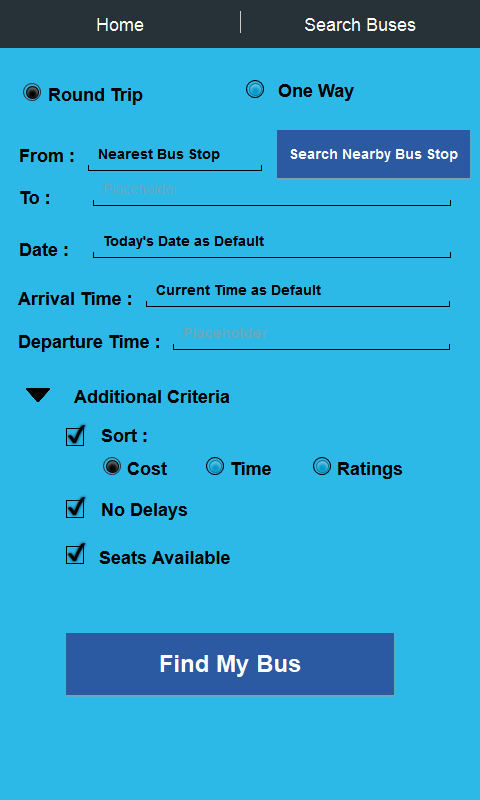
The wireframe UI for this view is built using online free proto.io designer. In order to view it online, visit - <https://pp231189.proto.io/share/?id=fbf04f22-70ee-4642-afea-1deee926bbe2&v=11> . There is also an offline storage of the App’s Wireframe UI being done which is located at the following dropbox link: <https://www.dropbox.com/s/wffz8s1yxey53j1/FindMyBusWireFrame.rar?dl=0> .

* 1. **Home Page View**

****

So the App is designed for users who wanted to search for Buses quickly using their Saved Search History. Hence, the App initially contains these two buttons – Find My Bus Using Saved Search Button and Find My Bus Using New Search Button rather than showing Search Screen directly.

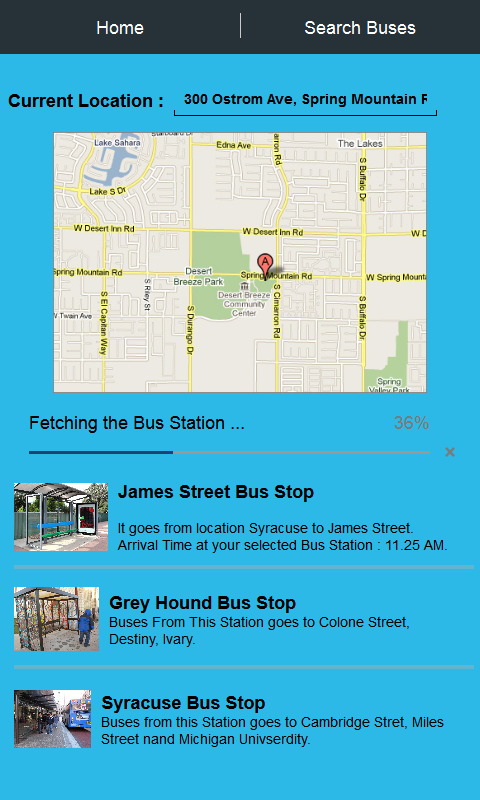
* 1. **Bus Search View**

****

It contains the Tab Navigation on Action Bar to go to the Home Screen directly. The next tab represents the same screen as Search Screen since the App analysis report projected this screen as the crucial screen where user would spend most of its time. This screen also serves as the controller for other screens like Bus Details and its sub-screens via Navigation Drawer, Nearest Bus Stop Screen and its sub-screens. For most of the future screens, these two tabs are expected to appear.

This Screen provides the functionality to search for the bus according to different criteria like type of Trip (one-way/ round trip), Nearest Bus stop, Date, Time, Arrival Time and Departure Time. It also provides additional optional criteria like sort by trip cost/ time/ ratings, No delays and Number of Seats Available.

* 1. **Bus Stop Search View (Nearest Bus Stop)**

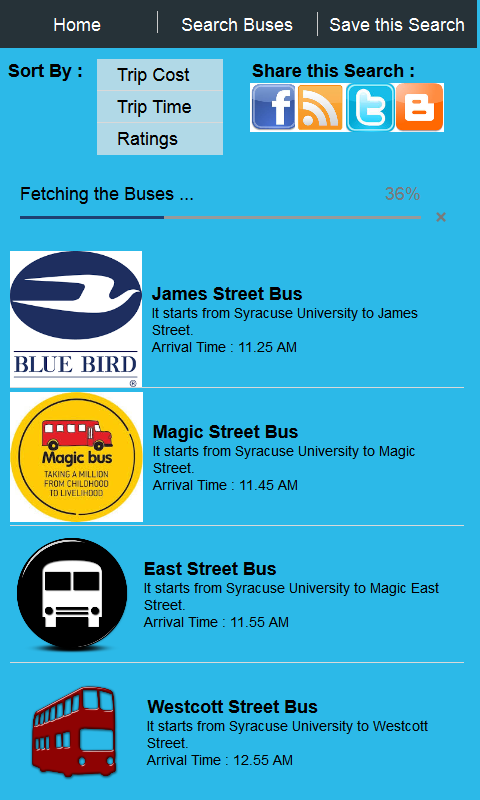
****

This screen is responsible for showing the bus stop or bus station nearest to your location. Also it has the functionality to change your location and find the corresponding bus station by providing the Google Map interface.

Since the data is to be presented by interacting different web service providers, some bus stops might not appear within expected amount of time. This is the reason for displaying progress bar by which user can get an idea about the time left for other results to show up.

Each item in the list of Near Bus station is clickable and will to Bus Stop Detail View Screen. This screen also contains the textbox in order to notify the current address to the user which the Google Map API is using to find the nearest bus stop.

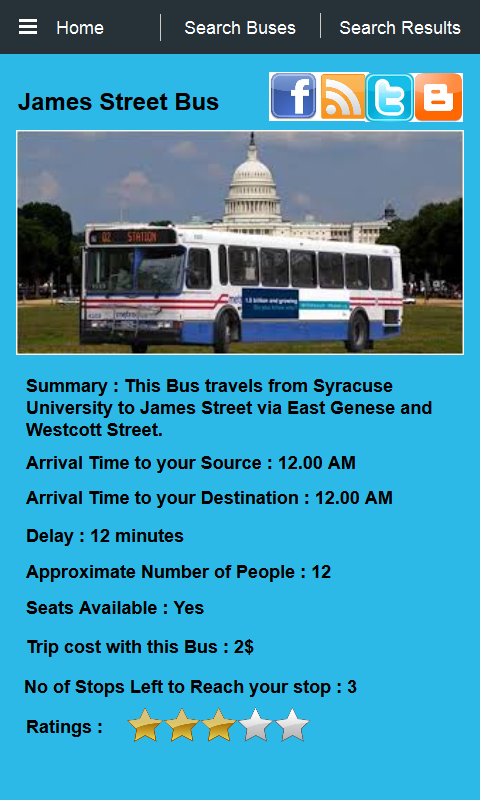
* 1. **Buses Listings View**

****

This is another important screen which can be thought as the controller of its sub-screens on Navigation Drawer. There is an additional tab in the Action Bar performing the functionality of saving this search. This functionality would save time of the users by not entering the search again and again.

There is a utility in the form of drop-down menu to sort the listings according to different criteria listed previously in the Search. The reason for including this sorting criteria option again because users are often expected to change their minds looking at the list and sort the current items via those optional criteria without using Back button or modifying the existing search.

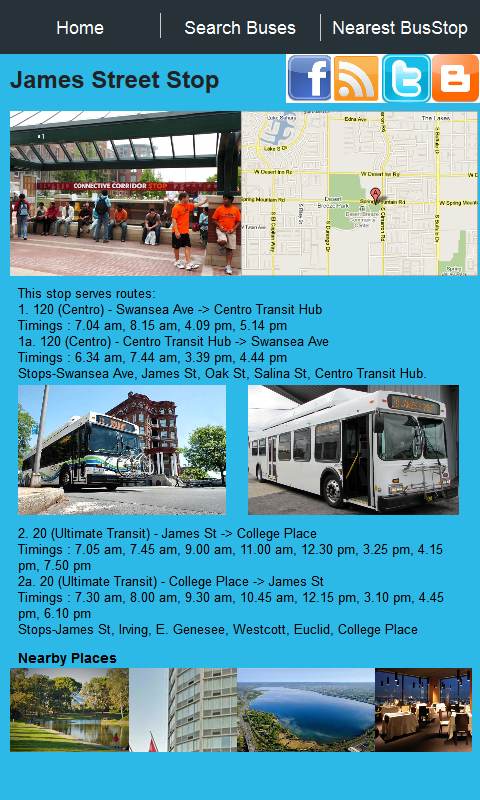
* 1. **Bus Detail View**

****

This screen have two extra components attached to the action bar – Navigation Drawer (to get more information and perform few actions regarding this bus) and Search Results Screen ( to get to previous screen). Users normally have a tendency to view details of multiple buses and wanted to navigate back to the previous. Providing the previous Screen Navigation in form of tab would encourage the user to explore more options. On clicking the leftmost burger icon, the navigation drawer shows up showing different options.

This screen provides the more details about the bus like its Summary about the journey, Arrival Time, Departure time, Delay, Approximate Number of People, Seats Available, Trip Cost, Number of Stops left and Ratings.

* 1. **Bus Stop Detail View**

****

This screen have an additional tab on the Action Bar which can navigate the user back to the listings screen of different nearby bus stops. Providing the previous Screen Navigation in form of tab would encourage the user to explore more options about the different nearby bus stops.

This screen contains the details about the bus stop or bus station like types of buses halting at the stop, timings of the halt, surrounding area pictures etc. It also acknowledges the user about other possible option of buses visiting to this particular stop.

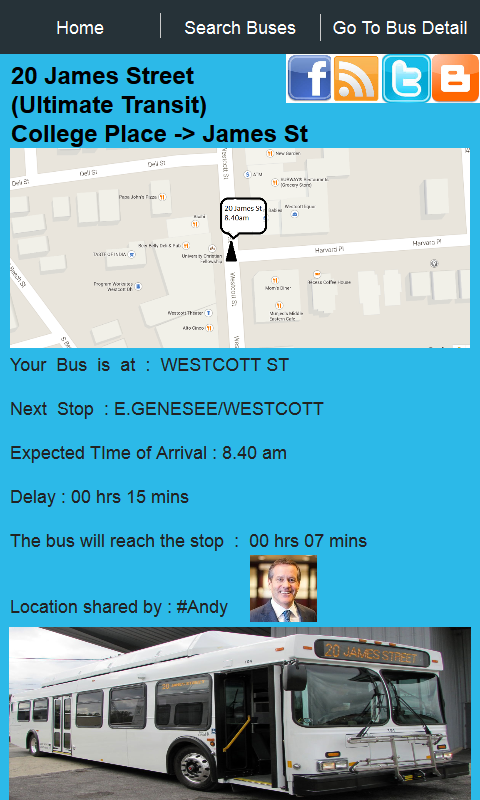
* 1. **Bus Detail Screen Drawer View**

****

This is the navigation drawer appeared after clicking on the burger icon in Bus Detail View Screen.

It has four different functionalities. “View Detail Schedule” will show up a new Fragment containing the entire Time Table of the bus. “Track My Bus” functionality allows the user to view the current or last saved location of the bus. “Wait For Me” functionality provides the ability to the user to ask the bus driver to wait for him/her. “Live Updates for this Bus” functionality will enable the option of viewing the feedback of other users who are boarding this bus currently.

* 1. **Track My Bus View**

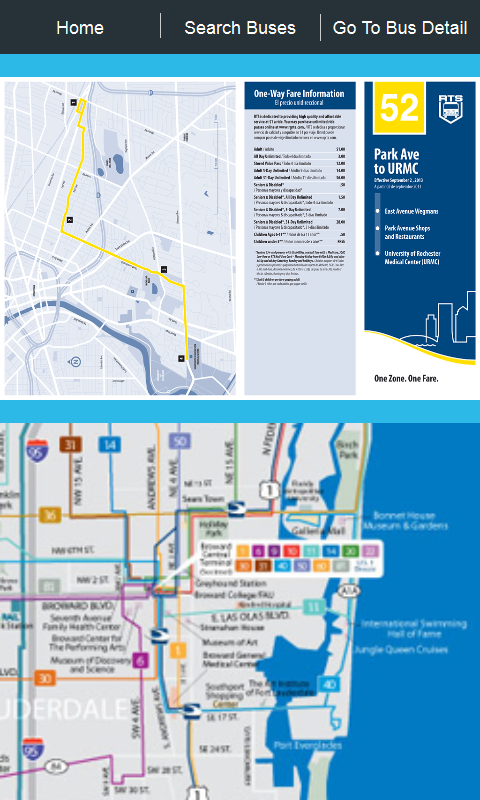
****

This screen have an additional tab on the Action Bar which can navigate the user back to the Bus Detail Screen. Providing the previous Screen Navigation in form of tab would encourage the user to tally its detailed functionalities with the summary and other minute details like ratings, number of stops, etc.

This is an exciting functionality which allows the user to track his selected bus by both options – by the online user’s feedback and by the real-time update. So with real-time update from GTFS web service, the exact location of the bus in the map is determined. Accordingly details like next stop, ETA and delay of the bus is calculated.

Additionally, you can also view the delay experienced about the bus posted by other users who are using this App and boarding on the same bus.

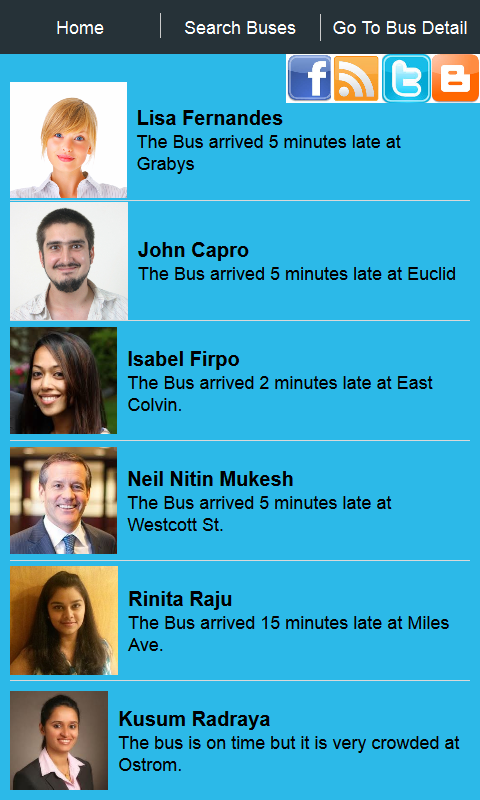
* 1. **Bus Detail Schedule View**

****

This screen have an additional tab on the Action Bar which can navigate the user back to the Bus Detail Screen. Providing the previous Screen Navigation in form of tab would encourage the user to tally its detailed functionalities with the summary and other minute details like ratings, number of stops, etc.

This screen provides the entire bus schedule in the pdf form so that the users can get themselves acknowledged about other halts of the current bus. It also provided the functionality to save this bus schedule details locally in pdf or word format.

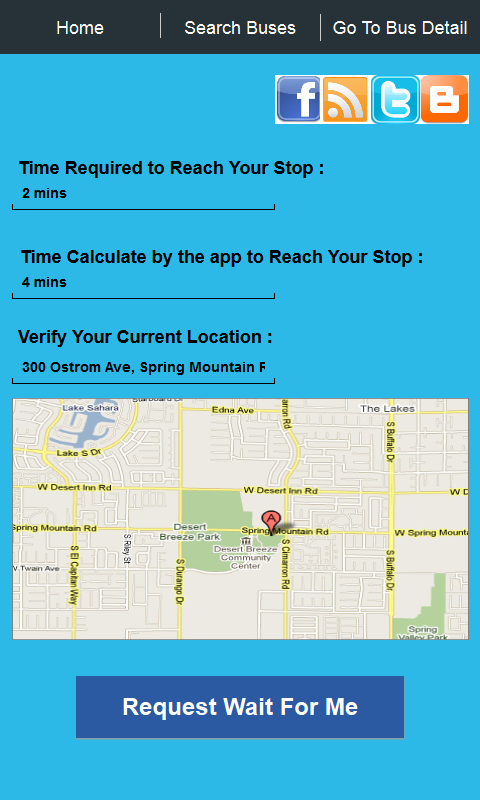
* 1. **Live Updates From Bus Users View**

****

This screen have an additional tab on the Action Bar which can navigate the user back to the Bus Detail Screen. Providing the previous Screen Navigation in form of tab would encourage the user to tally its detailed functionalities with the summary and other minute details like ratings, number of stops, etc.

This Screen is responsible for sharing the views regarding the bus amongst the users. Users of the app who are waiting 2 stops away and finds the bus status as crowded from the online status can decide to move on rather than wasting their on waiting for that bus. Additionally, users can also get themselves acknowledged about the bus delay at different stops so that they can start their journey from home a little late.

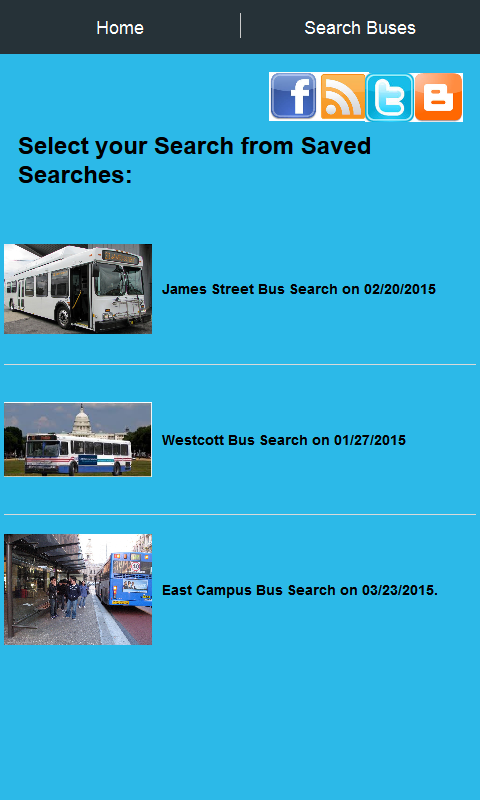
* 1. **Wait For Me View**

****

This screen have an additional tab on the Action Bar which can navigate the user back to the Bus Detail Screen. Providing the previous Screen Navigation in form of tab would encourage the user to tally its detailed functionalities with the summary and other minute details like ratings, number of stops, etc.

This is another new and exciting functionality which provides the opportunity to those users who are in real need of bus due to occasions like exams, interviews, etc. Such users often seemed to be in hasty mood and often miss the bus due to few seconds or around 1-2 minutes. There are bus drivers who show courtesy by waiting for such users. These bus drivers would get notification about the current user status. Depending on the wait time calculated by the app, the bus driver has the option to wait for such users who are just about to miss the bus for a minute. This Screen UI calculates the time required to reach the bus stop based on user’s current location. It also permits the user to enter his own inputs about time required if he/she is running towards the bus stop. The bus driver has the option to view the current status both ways – time estimation by the App and time estimation by the user.

* 1. **Saved Search View**

****

This screen provides the functionality to select the saved search from the Saved Search History. These Saved searches are expected to load very fast using the local app cache.

The reason for caching such searches because users use this functionality to save time from entering values in the Search screen.

If the App is loading these details from the database each time, then there seems to be some delay and overhead. Hence, a threshold is decided for the minimum number of searches to be saved in the local app cache depending on the cache space, processor etc.

* **Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.
  1. **Activity Handler**

Control first goes to the Main Activity Class. So this module is responsible for inflating different views and populate the necessary data obtained from the model into those views.

* 1. **Web Service Façade**

This module follows the Java Facade Design Pattern which is responsible for invoking different web services API and database related DAOs, doing necessary parsing and providing it to the Activity Handler Module. This module contains the business logic of the App.

* 1. **Adapter**

This module is used for setting different types of Adapters like ListViewAdatpers, RecyclerView Adpater, Fragment View Adapter, Custom Adapter, etc. into ViewLayout object. Custom Adapter is required to add a new functionality and providing utilities to use the existing API’s.

* 1. **View Adapter Proxy**

This module serves as the Proxy for Adapter Module. The code of setting different adapters to the view is contained in here. This module also adds another layer of abstraction to the Adapter Class by functioning like a Proxy.

**References:**

* MVC Pattern :[**http://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm**](http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm)
* MVC Pattern Sample Architecture Diagram :[**http://www.google.com/imgres?imgurl=http://www.tonymarston.net/php-mysql/model-view-controller-03.png&imgrefurl=http://www.tonymarston.net/php-mysql/model-view-controller.html&h=378&w=727&tbnid=7XD4F4F-k1kGQM:&zoom=1&docid=XeBKJr6C81lbHM&ei=mOcLVa76JoGzggSr-4PwAQ&tbm=isch&ved=0CDAQMygTMBM**](http://www.google.com/imgres?imgurl=http://www.tonymarston.net/php-mysql/model-view-controller-03.png&imgrefurl=http://www.tonymarston.net/php-mysql/model-view-controller.html&h=378&w=727&tbnid=7XD4F4F-k1kGQM:&zoom=1&docid=XeBKJr6C81lbHM&ei=mOcLVa76JoGzggSr-4PwAQ&tbm=isch&ved=0CDAQMygTMBM)
* GTFS :[**https://developers.google.com/transit/gtfs/**](https://developers.google.com/transit/gtfs/)