**JumpBus Design Specification**

**Overview**

JumpBus is a client/server solution that provides user-friendly timetable information to promote the best use of commuter time. The client application is is a mobile application, including a commuter version and a driver version, currently implemented on Android. The application server developed by the team uses data from the Bike*direct* network, Adelaide General Transit Feed Specification and Metro Real-time stop information. The server also collects useful statistics which might be used for service improvement.

**Solution Architecture**

Adelaide Metro Real-time Stop information

Bikedirect Network

Adelaide General Transit Feed

JumpBus Commuter App

JumpBus App Server

JumpBus Driver App

The JumpBus commuter version gets the data from three different data sets, and finally gives the user a best solution after a complicated calculation. JumpBus send notifications to bus driver via the JumpBus application server, from which the JumpBus Driver app is pulling requests. The JumpBus could also advise the real expect arrival time to Adelaide Metro Real-time Stop information (A future improvement).

**Application Feature**

1. User enters the home address, workplace address, the time they start work and are off work. Then the app will schedule the time for your departure and the bus you should choose everyday according to the real-time metro information data.
2. The app would provide the best solution to getting to the destination, the calculation the will involve the user’s current location, the specific time the user intent to get to the target place, the distance to the bus stops, the arrival time of the bus.
3. The app will advise the user to take another transportation if the current situation is unable to fulfill the user’s requirement (eg. Getting there in a limited time), probably the bike solution will pop up based on the bike direction data.
4. The user can accept a travel solution, then the alarm will be set to secure the commuter do not miss the bus, also the stop information will be sent to the server to notify the driver that a passenger will be waiting at the stop.
5. The app will give the bus information updates every 1 minute which could tell the user when the bus will arrive in real time. The user also can request the driver to wait for them maximum 2 minutes at stop if they are very likely to make it.
6. The bus arrived but the passenger could not make it, the waiting request will also be sent automatically when the background service determine the user can make to the bus stop in 2 minutes base on the current location and the walk pace.
7. The app for the driver’s side will get the waiting passenger information from the server every 1 minute. The bus driver will be notified how many passengers are waiting at the stop, this is most useful when it is in the night the driver may missed the people. It could also suggest the driver to wait for 1 or 2 minutes for those passengers who are very close to the stop rushing to getting on the bus.
8. The user also could rate or complain about the bus driver’s service when he/she gets on the bus. These information will be store on the server which could be a reference for driver’s performance assessment in the future.

**Further improvement**

1. All the computations are based on the real-time and dynamic data, eg the user location, bus location and walk pace.

2. The app would learn from the user’s travel habitat, which would be personalized data, eg. time to get up and go to bed, the place usually go, number of transfers has made to get to destination and how long it takes to get to the bus stop. These data could be uploaded to the server if the user wish, which could be a reference for transportation department to refine the bus routine services.

3. Providing ticket price information, and buying the ticket in the app before getting on the bus which could save the time for the bus driver to sell ticket to you, the ticket purchase validation is through NFC on the bus.

4. The driver also can update its traffic condition experience to make the real time data more precise since the lated expected arrival time is predicted based on algorithm.