Similar Apps Study for CS3281 Thematic Project – Crowd Motoring

1) Similar Apps Functions Statistics

	a Stop nearby	b Route Planning	c MRT map	d bus/stop bookmark	e bus notifi-cation	f search stop/ station	g Arrive time	h travel time
1	√	√		√	√	√	√	
2		√		√		√	√	
3		√		√	√	√	√	
4	√	√	√	√	√	√	√	
5	√	√	√	√	√	√	√	
6			√					√
7								√
	2	2	1	1	3	1	2	4
	<= 1	<=1	<1	<1	2	1	1	3

Above is a statistic table which records information about functionalities from similar apps. Row 1-7 indicates 7 similar apps, their names are included in the index of apps below. Column a to h represent different functionalities that various apps implemented, the detailed explanation of the functionalities and estimation about the time (in terms of weeks, recorded in the green row), requirement and difficulty (range from 1 to 5, 1 for easiest to develop and 5 for hardest to develop, recorded in the blue row) to develop such functions are included in the index of functionalities below.

- *Index of apps*

1. SBS Transit iris
2. SBS Next Bus
3. Singapore Bus Guide
4. Singapore MRT Information
5. LTA My Transport.SG
6. SG Trains
7. SMRT & LRT

- *Index of functionalities*

a. Stop nearby: According to the GPS location of the app user, the app displays information of stops that are within certain distance from the user (usually within 1km).
Estimated develop time: <= 1 week.

Estimated requirements: Google API
Estimated develop difficulty (1 for easiest, 5 for hardest): 2

b. Route planning: The app gets start location and destination location from user input, and then calculates the shortest/quickest route for traveling.
Estimated develop time: <= 1 week.
Estimated requirements: Google API
Estimated develop difficulty (1 for easiest, 5 for hardest): 2

c. MRT map: The app simply displays a map contains MRT lines in Singapore.
Estimated develop time: < 1 week.
Estimated requirements: an up-to-date MRT map of Singapore
Estimated develop difficulty (1 for easiest, 5 for hardest): 1

d. Stop/Bus bookmark: The app stores user's choice of favorite stop or/and bus, providing a shortcut for users to look up the bus/stop information conveniently.
Estimated develop time: < 1 week.
Estimated requirements: a schema for local data storage
Estimated develop difficulty (1 for easiest, 5 for hardest): 1

e. Bus notification: Based on the bookmarked bus, the app will notify user certain minutes before the bus comes.
Estimated develop time: 2 week.
Estimated requirements: knowledge for Android event handling; methods for Android notification functions
Estimated develop difficulty (1 for easiest, 5 for hardest): 3

f. Stop/Station search: The app returns information of bus stops/train stations (e.g. buses that pass by this stop, the stop id, etc.) when users try to search for a bus stop or train station.
Estimated develop time: 1 week.
Estimated requirements: database of bus stop and train station information
Estimated develop difficulty (1 for easiest, 5 for hardest): 2

g. Arrive time: The app displays the estimated arrive time for the next bus/train.
Estimated develop time: 1 week.
Estimated requirements: Google API
Estimated develop difficulty (1 for easiest, 5 for hardest): 2

h. Travel time: Given starting and ending stops, the app calculates the estimated (but to some extend accurate) time for a bus/train to get to the destination.
Estimated develop time: 3 week.
Estimated requirements: Google API, traffic data
Estimated develop difficulty (1 for easiest, 5 for hardest): 4

Analysis

1. Most implemented (most popular) features

- Stop/Bus bookmark
- Stop/Station search
- Route Planning
- Arrive time

2. Least implemented feature: Travel time
3. Easiest to implement features

- Stop/Bus bookmark
- Stop/Station search
- MRT map

4. Hardest to implement feature: Travel time

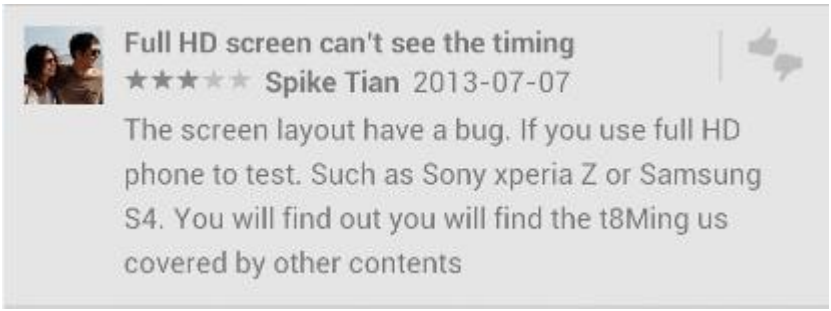
Conclusion

- Travel time is both the hardest to implement and the least popular feature, hence it is not likely to be implemented in our project;

- The most popular features, i.e. Stop/bus bookmark, stop/station search, route planning, and arrive time shall be implemented by us. Two of them (Stop/bus bookmark, stop/station search) are easiest to implement, which saves our time and resource. Another two might take some extra time and effort to do, but it is worthy;

- MRT map is also the easiest to implement, but it is not popular, and almost of no use. Hence we are not going to implement this feature.

2) User Feedback Analysis



The compatibility of our app on different android devices should be carefully considered. The design of the UI layout should allow devices with various sizes (such as smartphones and tablet) to display the UI correctly.

After upgrading, users can find difficulty using the app because of change of the layout or function modifications. It is essential to provide a user guide or change log for users to get used to new versions of the app.

Though bookmark function is widely implemented in similar apps, some users are not comfortable with its notification mode. Our app should provide some options on whether bookmarks will notify users about the buses they marked, or just provide a shortcut for them to look for those favorite buses. In this manner, the app will be less annoying and more user-friendly.

3. UI analysis

SG trains

Singapore MRT Information

LTA My Transport.SG

The 3 apps above have similar UI design: linear presentation of all functions. This is acceptable for apps with limited functions. However if an app has more functions, the UI layout will become very crowded and hard to click. If applying multiple pages to solve this problem, the flipping operation will make operating this app very inconvenient. Hence, in our app's UI design, a tab, a drop down list or whatever that can group functions while keep operations simple should be applied.