CPEN 321 Requirements Document

Group: Time Your Trip

Members: James Zhou, Johannes Gallmann,
Sirine Trigui, Yixin Zhao, Navjashan Singh

Requirements, Vision, and Scope

Product Vision Statement

Our product is a mobile application that sends alerts to the users when their bus is arriving at the user's chosen stop, to prevent them from missing it when doing other tasks on bus. The purpose of this app is to allow commuters to be more productive with their time during long commutes by allowing them to spend time doing useful tasks (ie. studying, sleeping) without having to worry about missing their stop. There are similar products which take users' destinations, and use GPS to get the locations, then alert users on designated stops. Our app will be similar to these apps, with the difference being that our app will offer offline information regarding bus routes, stops, and estimated arrival times in case of failure from the Translink Open API or if the user does not have internet connection.

Problem statement:

The problem of	Making long commute trips more productive for users
affects	TransLink bus users
The impact of which is	Instead of doing useful tasks, users will have to divert their attention to making sure that he/she gets off at the correct stop.
A successful solution would be	To come up with an app that will notify users when he/she reaches the chosen stop.

Product Position Statement:

for	Bus passengers
who	Provide us with some parameters as bus-stop # , bus # and destination stop. OR(select some required params from the provided lists)
Our system	Is a mobile application that utilizes the Android OS, GPS

	signal and the internet access capabilities of mobile phones.
that	prompts the user to enter the bus stop or select one, and select a bus number to return all the upcoming stops which the user has to select their final destination. By providing so, an estimated time will be returned and the user alarm will be turned on.
unlike	TravAlert; TranSit
Our product	Takes advantage of both real-time and static transit information and will switch between the two according to the status of the Translink Database and the user's own network status.

End users and feature list:

Translink bus passengers will be our end users. They will select some required parameters that the app will provide in different list views such as bus stop number, bus number and then their destination. Once they have entered all this information, the app will display the time it takes to reach the destination and will ask the user if they want to set an alarm to get alert before their destination arrives. The user is just expected to have some basic knowledge about how to input information into an app.

Constraints :

Our product works with the information received from TransLink using the TransLink OPEN API, Google Maps API, Google Distance Matrix API and and static text data that contains routes and stops information stored in a SQLite database. It's freely available and we can access by registering with them. As we are making an android app, our platform will be Android Studio.

We chose to use an embedded SQLite database for several reasons. One, it allows the user to access information offline, as SQLite is embedded into the app itself and does not require an external server to host information. Second, it is a relational database, and a lot of the data that the user needs to access are linked to one another (ie. stops times are linked to stops, which are in turn linked to a bus direction, which is in turn linked to a bus route). Thus, it's easier to construct queries that are based off of the user's requests through using the SQL format.

Scope and Limitations :

Our app will only be restricted to the Lower Mainland, as TransLink's database only covers this region. For this app to work as advertised, the user will require both internet access and GPS capability on his or her phone running the app. We will be using GPS data in order to track the user's current position and to use this factor to make our trip duration estimation algorithm more efficient and accurate. The app will store static Transit information in a SQLite embedded database, but this feature will not be able to provide real-time bus arrival estimates, and will instead only be based on estimated arrival times that TransLink posted.

Assumptions and Dependencies:

Our product is highly dependent on the Translink's API and several Google APIs, as it receives all the information regarding bus schedule, and the estimated arriving time from them. For our app to work properly, we will be expecting that the information we are receiving from TransLink and Google is correct and reliable. We will depend on the static text information released by TransLink for the offline feature to work correctly, as the static information will contain all the bus routes, directions, stops, and stop estimates that the user can access through the app.

Feature List:

1. Storing user inputs and formating a Query for the TransLink Open API based on them;
2. Map view at the launcher page for the user to see the bus stops around him/her in a 250 meter radius;
3. Display the location of every bus stop on the map;
4. Favorite list of bus stops and deleting/adding favorite stops;
5. Providing a helpful UI system that will guide users to input data;
6. Cancel feature that allows users to cancel inputs and start over;
7. Stop alarm feature for stopping the alarm;
8. Offline mode by using the embedded database;
9. Connecting to Translink Open API and retrieving relevant data in JSON format.
10. Parse JSON data to retrieve relevant information regarding arrival times.
11. Using GPS data to keep track of user's location relative to the location of the chosen destination stop.
12. Algorithm that calculates time of arrival based on Google API and GPS information.
13. Limiting phone data usage through minimizing the number of requests sent.
14. Accessing GTFS data whenever the Translink Open API is down, through developing queries that can access relevant data stored in the embedded SQLite database.

15. Set alarm/notifications based on the calculated estimated time of arrival (updating during the bus ride by frequently checking the current position)

User Stories and Use Cases :

Use Case: Set a notification for when the bus reaches the chosen stop

Identification:	Sets alarm based on real-time TransLink data for the user's bus
Primary Actor	A commuter about to board a bus.
Stakeholders & Interests	Commuters would want to take advantage of this use case in order to maximize their efficiency while riding the bus.
Preconditions	● TransLink Database and Google API are operational ● App is currently displaying starting page ● No inputs have been registered ● User must have access to a bus stop number ● Device must have access to internet ● Device must have access to GPS data
Postconditions	Success: ● An alarm/notification is set ● A window is displayed with a "Reset" button and a Stop Alarm" button. The former one navigates to the starting window app, allowing the user to start over and cancel the previously set alarm; the later one stops the alarm. Failure: ● An error toast is displayed ● App resets to its starting menu with the previous parameters wiped.
Main Success Scenario	1. User inputs or selects the bus stop number 2. System verifies and displays the buses arriving in the next 30 minutes (sorted list starting with the first bus arriving at that stop). 3. User selects a bus from display.. 4. System verifies and displays all of the upcoming stops for that bus 5. User selects a stop. 6. System verifies, calculates, and sets the timer to ring when the user reaches that stop.
Extensions & Alternative Flows	1. Bus stop number cannot be identified (error): ● Error toast is displayed ● System returns to starting menu, prompting user to input bus stop

	number again. 2 .No busses can be detected within the time frame (error) ● System displays the first bus that will be arriving at the selected stop 3. User presses Stop Alarm button ● The alarm will stop ringing. 4. User presses Reset button ● System resets to the main menu, the set timer is canceled, and the system redirects to the starting menu with all inputs wiped.
Open Issues	● When the TransLink database is offline. ● When the user does not have internet access. Hypothetical solutions: default to stored GTFS database.

Use Case 2: Provide offline bus arrival estimated times

Identification:	Provides estimated bus arrival times for every stop offline
Primary Actor	A commuter waiting at a stop or wanting to get off at a stop
Stakeholders & Interests	Commuters often do not have internet access, and having an offline TransLink reference will be extremely beneficial for on-the-fly trip planning.
Preconditions	● The most recent GTFS data is downloaded and stored in the embedded SQLite database
Postconditions	Success: ● A page that lists the information for a certain stop along with all the estimated arrival times for all buses for this stop in the next 30 minutes (or the next available bus if buses are arriving outside of the 30 minutes timeframe) Failure: ● An error toast is displayed ● App resets to its starting menu with the previous parameters wiped.
Main Success Scenario	1. User selects a bus route 2. System verifies and displays all the directions for that route 3. User chooses a direction 4. System verifies and displays all of the stops for that that direction 5. User selects stop. 6. System verifies and displays the estimated arrival times for the selected stop
Extensions & Alternative Flows	1a. Route number cannot be identified (error): ● Error toast is displayed ● System returns to starting menu, prompting user to input route number

	again 2a .No busses can be detected within the time frame (error) ● System displays the first bus that will be arriving at the selected stop 2b. User inputs a select bus ID 1. Bus ID cannot be identified -System stays on current page, toasts that bus ID is invalid, and prompts user to choose a bus displayed or to input an ID again 2. Bus ID is identified. -System proceeds with from step 4 2c. User presses undo button ● System resets to main menu and the previous inputs are wiped. 3. User presses undo button ● System resets to the bus display menu and the previous input for bus selection is wiped
Open issues	GTFS data only provides estimated times, does not produce real-time transit information

Use Case 3: Displays the user's current location on a map and place markers for bus stops within the user's 250 meter radius.

Identification:	Access GPS coordinates for bus stops from TransLink database and retrieve the ones closest to the user's current GPS location
Primary Actor	A commuter unsure of which bus stops are nearby
Stakeholders & Interests	Commuters would want to use this feature if he/she does not remember the bus stop number for a stop that is nearby.
Preconditions	● TransLink Database and Google API are operational ● App is currently displaying starting page ● No inputs have been registered ● Device must have access to internet ● Device must have access to GPS data
Postconditions	Success: ● A set of markers is displayed near the user's current location. ● A button labeled "View My Location" from the sliding menu will allow the user to refresh his/her current location and refresh any markers that are changed Failure: ● (If no GPS data) The map will not zoom to user's current location ● (If no Internet or if no stops nearby) the map will not display markers

	near the user's location
Main Success Scenario	1. User loads the application (or click on the View My Location button in the sliding menu) 2. The app will locate the user's current position and will zoom towards it on the map, where the individual street names and building names will be visible 3. A set of markers will be placed on the map that will indicated valid bus stops. The user can click on a marker and it will display the bus stop number, bus stop name. A second click will enable the app to display a new activity containing a list of buses arriving (as in use case 1)
Extensions & Alternative Flows	1. User's location cannot be identified ● Error toast is displayed ● Map will not zoom to the user's location. The search view functionality (entering a stop number) will still work. 2 .No busses can be detected within the radius ● Map will zoom to user's location, but no markers will be placed and a message will be displayed indicating no stops around. 3. User presses View My Location ● The user's location will be refreshed and new bus stop markers will be added based on the user's new location 4. No Internet access ● System zooms to user's location, but no markers will be displayed and an error message will indicate no network access.
Issues	● When the TransLink database is offline. ● When the user does not have internet access. Solutions: keep map static and un-interactive. Default to use case 2.

Non Functional Requirements:

Fortunately our app does not have any safety requirements, since there is no way that the use of the app results in loss, damage or harm. The worst thing that can happen is that the alarm would not go off and the user misses his destination stop. Provided that the GPS is working properly this should be achieved.

Because we're not storing user data on the device or sending it through the network we have no security requirements for this app either. At no point of time the user has to provide personal data so it is all anonymous.

Security Requirements

Based on the working of this app, it asks for user's starting stop and the destination stop and will set up an alarm based on the time of the journey. It doesn't require any authentication or any personal information from the user apart from the GPS location from the user which we are using to set up and trigger the alarm when the user is in range of 300 metres from his destination. For security and location privacy reasons, we are only using Google's time calculating feature based on two GPS coordinates so as to get an estimate remaining journey distance. As we are only sending a query based on the GPS coordinates and updating the remaining time, it will not hinder with the user's security.

Software Quality Attributes

Adaptability

Our app adapts according to the environment as it checks the alarm time after reaching half of the initial estimated time to get a better estimate of the time remaining. It enables the app to adjust the time based on external factors such as traffic, accidents or other issues that can change the estimated arrival time from the ideal time. Also, as the app will also have an offline version where we will not be using internet connection and we will be relying on the information in the database. For that, we will be updating our app whenever the expected timing schedule of translink changes and our app will updated accordingly.

Availability

The app requires internet connection and location services for its optimum functioning and will be relying on the stored database for the offline working. The app will be available to use at all times but as the transit service shuts down for some buses late at night, so our app will display the timings for the arrival for the next bus even though it might be in the next morning.

Correctness

Our app has a distance checking algorithm which checks the time remaining after half of the initial estimated time has passed and updates with a new estimated time. Also, when the mobile is in range of 1000 metres of the destination, then the GPS sends continuous location updates and will sound the alarm when the user is within 300 metres of radius from the destination. By implementing this algorithm, we are aiming to achieve as correct timings and results as possible.

Interoperability

Our app is dependent on Translink's API, Google API and GTFS for receiving the required information and processing it to generate the estimated arrival time of the app's user. We do not have any restrictions on the data received from the Google API but we do have a limit of

requests for Translink API. Apart from that, as GTFS is a static database which is on the user's phone, it is possible to access that data without any restrictions.

Maintainability

For the maintenance of the app, our primary focus will be the user's feedback as we will keep fixing the bugs which may come up when the app runs in the real world. Apart from that, we will be looking for improving or implementing features to improve the user experience in the app. Additionally, as the app will have an offline version also for which we will be using GTFS, a static database to store the trip information, we will be updating and maintaining it depending on the changes of timings by the translink.

Portability

As it is an android app, any android user who has Android version greater than or equal to 4 can install it on his/her android phone. As the app works on an android phone, it is portable and can be used while travelling in a bus in lower mainland to compute their trip time.

Testability

We have been testing the testable elements of our app using testing platforms that android provides such as unit testing, instrumentation testing etcetera. Apart from that, we are also testing our app thoroughly in real world by travelling in buses and checking how our app responds in order to catch the bugs that may arise. As we have already released a beta version of our app on the google play store, we are also monitoring and working on the feedback that we are getting from the consumers of our app.