

## **Bus Tracker**

**A bus tracking app optimized for CMU students**

# **User Requirements Document Project Submission**

### **Prepared by:**

Daniel Stoll  
Ryler Hockenbury  
Evgeny Toropov

# Table of Contents

## [I. System Overview](#)

[Motivation](#)

[Existing Systems](#)

[Goal](#)

[Target User](#)

[High Level Features](#)

## [II. Requirements](#)

## [III. Features](#)

## [IV. Use Cases](#)

[Use Case 1: Find Next Bus](#)

[Use Case 2: Get up-to-date info](#)

[Use Case 3: Update server database](#)

[Use Case 4: Update server configurations](#)

## [V. UI Illustrations](#)

[Loading Screen](#)

[Home Page](#)

[Locate Stop Screen](#)

[Lookup Stop by Address Screen](#)

[Schedule Screen](#)

[Sequential UI Flow](#)

## [VI. Data](#)

## [VII. References](#)

## [VIII. Business Model Canvas](#)

# I. System Overview

## Motivation

There is a big problem for CMU students who need to locate bus routes and stops quickly and easily. New students often do not know which bus to take when attempting to travel places in Pittsburgh, and the information they need is not in one place. Pittsburgh Port Authority busses operated by the City of Pittsburgh does not coordinate with the CMU Shuttle service to offer a collective, unified interface to these bus systems for students. The main goal of our app is to let users access Pittsburgh Port Authority bus information *and* CMU shuttle service information all in one place. Students would be able to use this app to search what time a bus would arrive at a particular bus stop.

## Existing Systems

There are two relevant currently deployed bus information services: the CMU shuttle service and the Port Authority bus service. The CMU shuttle service has a real-time bus tracker [webpage](#) and mobile app (AndysBuses). This is useful because it displays the current direction of the bus, whether it is stopped or moving, and its route. The Port Authority bus service does not have a published mobile app, although they provide a list of third party apps which allow a person to navigate the Pittsburgh bus system. Instead they have a mobile website, but this site is similar to the desktop [site](#) and requires a large amount of text entry to do searches. Currently, the Port Authority is testing a real-time bus [tracking system](#) (on one bus route) but this will not be deployed for all their buses until sometime in 2014. Please see References section for more links.

## Goal

To provide a way to see the arrival time of the next bus with a few clicks.

## Target User

The target audience for the bus tracker app are CMU students. A student may use the app multiple times per day to locate bus stations and bus schedules. Only the bus routes and schedules most relevant to CMU students will be implemented in the app to optimize its usefulness for CMU students. This will include all the CMU shuttle schedules and routes, and the Port Authority bus schedules and routes that stop at any of the CMU bus stations.

## Key Features

- **Simple interface:** Bus lookup needs to be streamlined by minimizing input. Any text entry will try to be avoided so that the interface is easy to use in the colder weather when a user may be wearing gloves. Motions and gestures will be included to make interacting with the app even simpler.
- **Favorites menu:** Users are able to add stops to their favorites. Favorite stops appear first when the user views a list of the stops. This will allow faster access to the stops that are most used.
- **Off-line mode:** The app can be used in an off-line mode once an initial connection is made with the server. The schedules may not be synced with the server, but this is acceptable if the schedules at the server have not changed since the last connection. The app indicates to a user whether data from the phone or from the server was used.
- **Targeted schedules:** The app provides users with the most relevant bus stops and bus route information. This includes the CMU schedules and the Port Authority buses that travel to the CMU campus. Including only a subset of all the Pittsburgh buses saves memory resources on the phone, speeds up the app, reduces time spent connecting to the server, and unclutters the interface.
- **Real-time data (*in future*):** CMU shuttles have a gps tracking system, gps tracking of Pittsburgh buses is at development.  
Unfortunately, we were not able to contact either andysbuses.com (manages CMU shuttle tracking system) or cleverDevices (manages PAT buses tracking system). Any help in this organizational issue would be appreciated.

## II. Requirements

1. The system maintains accurate information by accessing schedule information available from the CMU shuttle service website and the Port Authority website.
2. The system contains information on both CMU shuttle service and Port Authority buses. This information may include stops, schedules, and current location for each bus route.
3. The system uses a webservice to provide bus schedule updates to the app when the app is connected to the internet.
4. The bus lookup services are available via app and web-interface.
5. The system supports the following utilities:
  - a. Bus stop lookup - a user may find a listing of all bus stops listed by distance from user, or from a specified address
  - b. Closest bus stop lookup - by default, the closest bus station to the users current location is displayed, and the results of a bus station search are sorted by distance from the user
  - c. Favorites stops - a user can add a stop to favorites
6. Potential features include real time bus tracking - Port Authority transportation system is currently testing GPS-based bus tracking.

### III. Required features

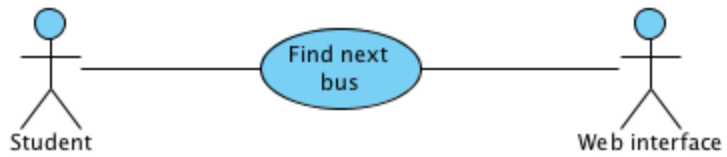
1. **Hardware Audio:** The audio can be used to give the user feedback when they are pressing buttons, or selecting radio or check boxes. A click sound will indicate that the selection or press was successful, and a soft thud can indicate the target cannot be selected or pressed. Sounds are also generated for the swipe action and the shake motion.
2. **Location:** Needs the user's current location to locate the nearest bus stops.
3. **Network-based Geo location:** Retrieves the user's current location from the service provider when GPS cannot be accessed
4. **GPS:** Retrieves the users location based on GPS
5. **Accelerometer:** Is used to detect shaking a device. In that case, the times are updated.
6. **Motions:** The app updates the arrival times list for the buses when the phone is shaken (i.e. shake to update).
7. **Gestures:** A touch will be used to select or press an item on the screen. A long press may be used to paste an address into the address bar for a bus stop lookup. Swipe will be used to move back a screen. If the optional map based ideas are implemented, the pinch open, pinch close and double touch gestures will be available for zooming on the map.
8. **Touchscreen:** The touchscreen will enable the user to interact with the app by tapping and swiping. The user will also need to enter data in a text field by tapping on the on-screen keyboard on some screens.
9. **SQL Lite DB:** The database at our server will store bus stop locations and the bus schedules that is acquired through the Port Authority and/or CMU shuttle services webpages. It is expected to contain 3-4 tables, although the specifics, associations, and keys have not been decided on. The server will query this database to satisfy requests from the app.
10. **Web Services:** The app can access the bus schedules at our server in order to download and store the information. Web interface to bus lookup is available.

## IV. Use Cases

Use case 1:  
Find next bus in app



Use case 2:  
Find next bus in browser



Use case 3:  
Update local schedule



Use case 4:  
Add route to favorites



## Use Case 1: Find next bus in app

Use Case Name:	Find next bus in app
Actor(s):	Student, App
Description:	A student gets the arrival times for next several buses using an Android device
Pre-conditions:	The app is open and on the home screen
Normal flow of events:	<ol style="list-style-type: none"><li>1. App offers to choose the bus stop</li><li>2. Student chooses the bus stop either by distance or by street</li><li>3. Student chooses buses coming through that stop and their direction</li><li>4. App lists the upcoming buses for the specified bus stop, and indicates if local or remote data source was used</li><li>5. App updates arrival times if the phone is shaken</li></ol>
Post-conditions:	A list of buses with corresponding arrival times is displayed for the stop selected.
Frequency of usage	High
Alternative flow of events:	<ol style="list-style-type: none"><li>1. No GPS signal and no network-location data =&gt; The choice of the bus stop based on distance is disabled</li><li>2. No internet connection =&gt; Schedule update and real-time bus locations lookup is disabled.</li></ol>
Exceptions:	There is internet, but the connection to Server cannot be established => report an error and proceed without real-time bus locations lookup
Assumptions:	Student knows what bus routes he/she needs
Issues:	None
Source:	Requirements list
<b>Hardware features:</b>	GPS and network-based location for finding the closest bus stop; Touchscreen for interface Accelerometer to trigger the arrival times update

## Use Case 2: Find next bus in browser



Use Case Name:	Find next bus in browser
Actor(s):	Student, Web interface
Description:	A student gets the arrival times for next several buses via web browser
Pre-conditions:	A web browser is open
Normal flow of events:	<ol style="list-style-type: none"> <li>1. Student opens the web interface home page</li> <li>2. Student chooses by street</li> <li>3. Student chooses buses coming through that stop and their direction</li> <li>4. App lists the upcoming buses for the specified bus stop</li> </ol>
Post-conditions:	A list of buses with corresponding arrival times is displayed for the stop selected in the browser
Frequency of usage	High
Alternative flow of events:	
Exceptions:	
Assumptions:	Student knows what bus routes he/she needs
Issues:	None
Source:	Requirements list
<b>Hardware features:</b>	Web service

### Use Case 3: Update local schedule

Use Case Name:	Update local schedule
Actor(s):	Student, App
Description:	The student updates the schedule stored on the device in order to be able to get the schedule without network connection
Pre-conditions:	There is internet connection, the home page on the app is open
Normal flow of events:	1. User opens menu on the home page and chooses "update phone schedule" entry 2. The user waits until the schedule is updated and the corresponding message appears
Post-conditions:	Phone database is updated
Frequency of usage	Low
Alternative flow of events:	
Exceptions:	There is internet, but the connection to Server cannot be established => report an error and ask to try later
Assumptions:	Administrator has a valid username + password
Issues:	None
Source:	Requirements list
<b>Hardware features:</b>	Webservice Network

#### Use Case 4: Save a stop to favorites

Use Case Name:	Add stop to favorites
Actor(s):	Student, App
Description:	A student saves the selected stop in order access it faster the next time.
Pre-conditions:	None
Normal flow of events:	1. App offers to choose the bus stop 2. Student chooses the bus stop either by distance or by street 3. Student opens menu and chooses "add to favorites"
Post-conditions:	Favorite stops are updated, next time when the student searches for the stop, the selected favorite will be in the beginning of the list
Frequency of usage	Low
Alternative flow of events:	Bus stop is already in the favorites => the corresponding message is displayed
Exceptions:	
Assumptions:	Administrator has a valid username + password
Issues:	None
Source:	Requirements list
<b>Hardware features:</b>	Webservice for changing data on server

## V. UI Illustrations

### Home Screen

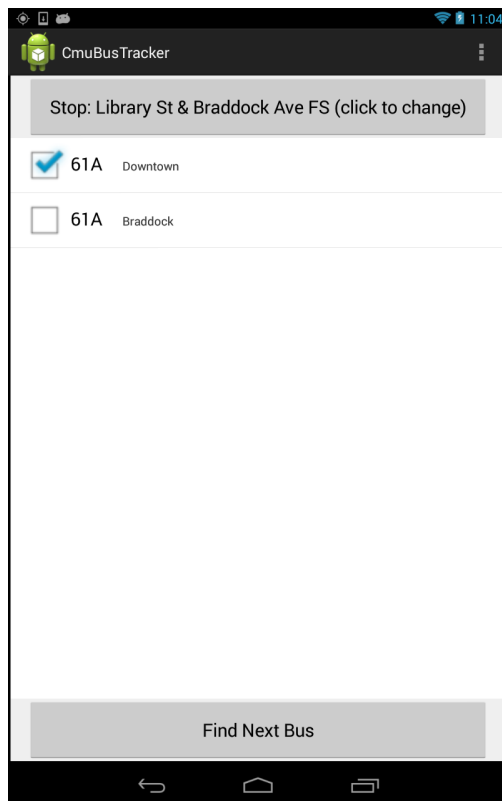
The page which is displayed after the loading screen. This screen gives the user the ability to select a bus station (the default is the closest station to the user's current location), select a bus by number, and choose whether the user wants to travel to or from the CMU campus. The list of buses is dynamically populated by only the buses that pass through the chosen station. By specifying the station, bus and direction the user will be taken to the schedule page after pressing the "Find Next Bus" button.

If the user wishes to change the bus station, he/she may click on the current station to switch to the locate stop screen to select a new station.

The context menu contains "Add to favorites" (see Locate Stop Screen), "Update database", "Settings", and "Help" items. When calling "Help", a dialog with instructions appears.

**Gestures:** Touch to select radio buttons and buttons.

**Features:** Audio for sound feedback, location to find user's current location, gestures to interact with screen, touchscreen, webservice



## Locate Stop Screen

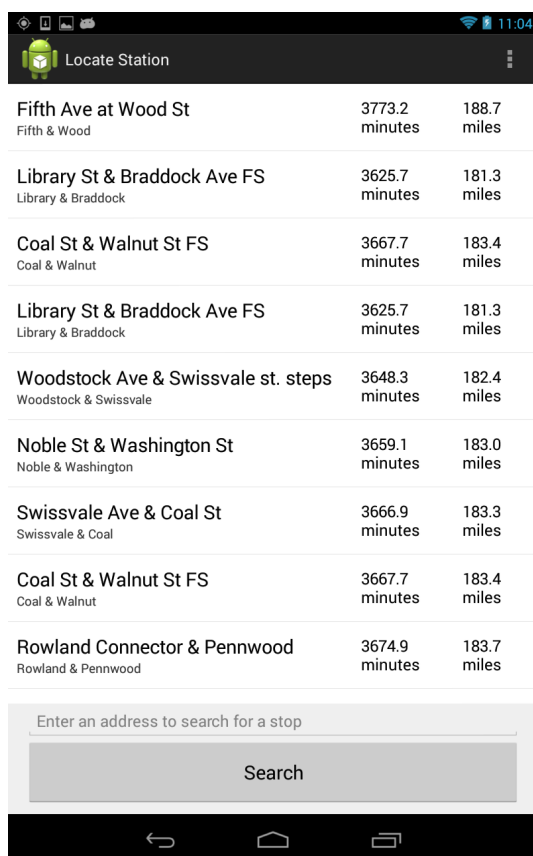
This page lists bus stops sorted by distance from the user. The stops added to favorites (see Home Screen) are always displayed in the beginning of the list. The user may select the stop by pressing on the tiles, which will take the user back to the home screen to select a bus and direction.

Additionally, the user may choose to search for a different station by entering an address in the text field and pressing the “lookup station” button. This will switch the screen to the lookup stop by address screen.

The context menu contains “Help” item, which opens a dialog with instructions.

**Gestures:** Touch to select a bus station, or select the touch entry field. Long press to open action bar to paste an address on clipboard. Swipe to move back to home screen.

**Features:** audio for sound feedback, location to find user’s current location, motions to shake to update bus station list, gestures, touchscreen, webservises



## Lookup Stop by Address Screen

This screen will display a list of stations sorted by distance from the address entered.

“Address” is one street or an intersection of two streets. Only the stations where buses from selected routes stop are displayed. The user may select a station by clicking on the tile, which will take the user to the home screen to then select a bus and direction.

The context menu contains “Help” item, which opens a dialog with instructions.

**Gestures:** Touch to select a bus station. Swipe to move back to locate stop screen.

**Features:** audio for sound feedback, gestures, touchscreen, webservices

Station Name	Address	Time (minutes)	Distance (miles)
Forbes Ave at East End	Forbes & East End	3682.6	184.1
Forbes Ave at Braddock St FS	Forbes & Braddock	3683.5	184.2
Forbes Ave btw. Dennison & Shady	Forbes & Shady	3700.8	185.0
Forbes Ave at Shady St	Forbes & Shady	3701.2	185.1
Forbes Ave at Murray St	Forbes & Murray	3704.9	185.2
Forbes Ave & South Craig St	Forbes & South Craig	3731.6	186.6
Forbes Ave opposite Craig St	Forbes & Craig	3731.7	186.6
Forbes Ave at Atwood St	Forbes & Atwood	3736.9	186.8

Help

Update Schedule

Settings

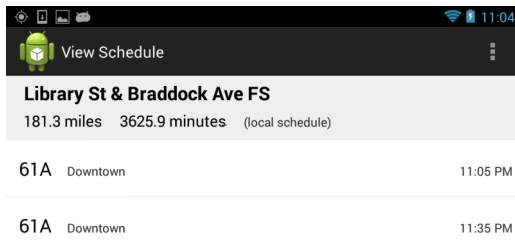
Add to Favorites

## Schedule Screen

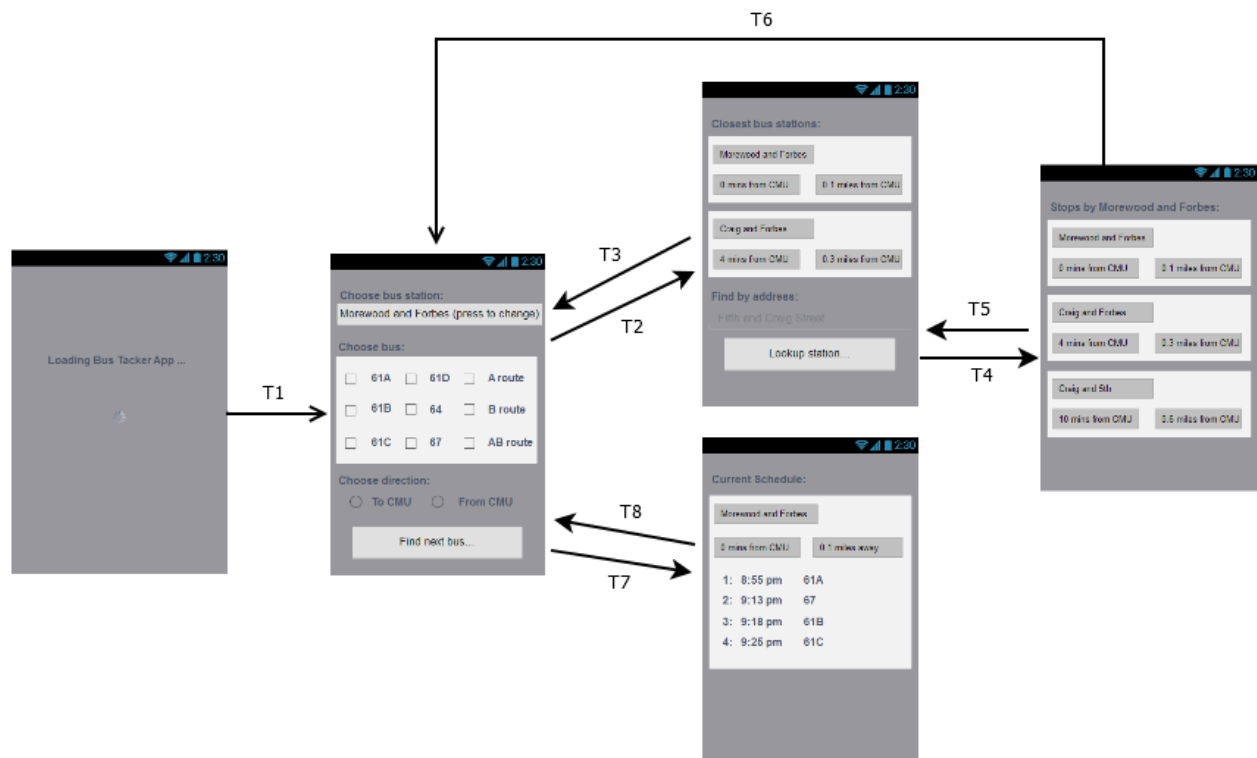
This screen shows the current schedule for the chosen station and buses. The list of buses begins with the next stop of the bus selected on the home screen.

**Gestures:** Swipe to move back to home screen.

**Features:** motions to update bus arrival list, gestures, touchscreen, webservices



## Sequential UI Flow



Transition	Action
T1	No action, app loads
T2	User taps “choose bus station” button
T3	User selects a new bus stop from the sorted list of stops, swipes back, or presses phone “back” button, sends station id in bundle
T4	User enters or pastes an address into the text field and taps “lookup station” button, sends text search in bundle
T5	User swipes back or presses phone “back” button
T6	User selects a new bus stop from the sorted list of stops, sends station id in bundle
T7	User selects at least one bus and a direction, and taps “find next bus”, sends stop and buses in bundle
T8	User swipes back or presses phone “back” button



## VI. Data

There are 3 possible sources of data for every request

1. schedule on phone
2. schedule on server
3. real time data from PAT bus and CMU shuttle systems (not implemented)

When the network is not available, the app uses local schedule. When network is available, it sends a query to the server. The server decides, which data source to use for the stop/bus/direction combination.

Unfortunately, we were not able to contact either andysbuses.com (manages CMU shuttle tracking system) or cleverDevices (manages PAT buses tracking system). Any help in this organizational issue would be appreciated.

Once in a while a user should update local schedule. He does that tapping “Update schedule” item in the Home Screen context menu. In this case the local database is updated from the server.

## VII. References

Tiramisu - Pittsburgh real-time bus tracker project. Based on having people report their GPS coordinates while riding buses: <http://www.tiramisutransit.com/>

CMU Shuttle Website: <http://www.cmu.edu/police/shuttleandescort/>

CMU Shuttle Tracker Website: <http://www.andysbuses.com/>

Port Authority Website: <http://www.portauthority.org/paac/default.aspx>

Port Authority Tracker Website (currently in testing):

<http://74.116.73.3/bustime/map/displaymap.jsp>

Relevant Student Project: <https://sites.google.com/site/pittsburghportauthorityapp/>

Port Authority Mobile Website: <http://www.portauthority.org/mobile/>

List of Third Party Bus Tracking Apps:

<http://www.portauthority.org/paac/RiderServices/GoMobile.aspx>

## VIII. Business Model Canvas

