

Bus Tracker

A bus tracking app optimized for CMU students

User Requirements Document Unit 2

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. References](#)

[VII. 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.

High Level Features

- **Quick Lookup:** Bus lookup needs to be streamlined by minimizing input. The goal is to push relevant information (i.e. the closest bus / bus station) to the user as defaults rather than have the user parse lists of busses and stations.
- **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 have not changed since the last connection to the server.
- **Status Indicator:** A status indicator bar will inform the user if the currently displayed schedules are up to date (meaning they are synchronized with the schedules on the server). The indicator can show “up to date”, “updating”, “out of date”, or “unable to update”. This will indicate to the user if they are running in an off-line mode.
- **Simple Interface:** The app will have a minimalistic interface designed to easily reach the schedules and bus stop information with only a few input fields. 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.
- **Targeted Schedules:** The app will provide 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. By including only a subset of all the Pittsburgh buses, it will save memory resources on the phone, speed up the app, reduce time spent connecting to the server, and unclutter the interface.

II. Requirements

1. The system will maintain accurate information by accessing schedule information available from the CMU shuttle service website and the Port Authority website.
2. The system will contain 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 will use a webservice to provide bus schedule updates to the app when the app is connected to the internet.
4. The system will support 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

5. Potential features include (time permitting):
 - a. Map lookup - a user may search for a bus route or bus stop by looking at a map overlaid with all the bus routes and bus stops
 - b. Real time bus tracking - current location of bus overlaid on Google maps interface
6. The system will support role-based access for the following user types:
 - a. User - a student may obtain route information for Pittsburgh Port Authority buses and CMU shuttle service
 - b. Administrator - an administrator may add, delete or modify data base on the system server

III. 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.
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. **MultiWindow:** Enables the user to view our app at the same time as another app in the same window.
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 needs to access the bus schedules at our server in order to download and store the information.

IV. Use Cases

Visual Paradigm for UML Community Edition [not for commercial use]

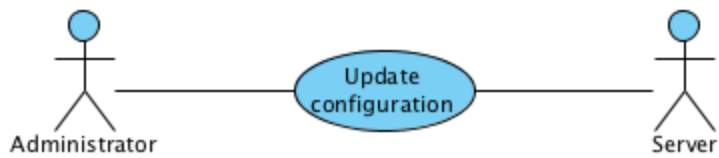
Next bus



Get up-to-date info



Update configurations



Update databases



Use Case 1: Find Next Bus

Use Case Name:	Find next bus
Actor(s):	Student (a student in Pittsburgh CMU campus), App
Description:	User selects the buses he needs and the bus stop, and gets the arrival times for next several buses
Pre-conditions:	The app is open and on the home screen.
Normal flow of events:	<ol style="list-style-type: none">1. Student selects several bus routes from the list and direction2. App offers to choose the bus stop3. Student chooses the bus stop either by distance or by street4. App lists the upcoming buses for the specified bus stop.5. App updates arrival times if the phone is shaken
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">1. No GPS signal and no network-location data => The choice of the bus stop based on distance is disabled2. No internet connection => Schedule update and real-time bus locations lookup is disabled.3. Local database is empty, but the connection to Server cannot be established => "Try again later" message
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
Hardware features:	GPS and network-based location for finding the closest bus stop; Touchscreen for interface Accelerometer to trigger the arrival times update

Use Case 2: Get up-to-date info

Use Case Name:	Get up-to-date info
Actor(s):	Student (a student in Pittsburgh CMU campus), App
Description:	Student updates local database from the server
Pre-conditions:	There is internet connection
Normal flow of events:	1. Student chooses to do database update 2. App performs update, App status bar indicates this process 3. App finished update, App status bar indicates success
Post-conditions:	Local databases are updated
Frequency of usage	High
Alternative flow of events:	System could not connect to the server or error during connection => The user is asked to try later
Exceptions:	
Assumptions:	None
Issues:	None
Source:	Requirements list
Hardware features:	Touchscreen for interface Network connection or wi-fi for connecting to our server Hardware audio for update notification Webservice for accessing data

Use Case 3: Update server database

Use Case Name:	Update server database
Actor(s):	Service administrator, Server
Description:	Administrator logs in to the server and updates information on bus routes, stops addresses, and bus schedule
Pre-conditions:	All databases already exist on the server
Normal flow of events:	<ol style="list-style-type: none">1. Administrator logs in to the server.2. Administrator selects to update a certain database3. Administrator performs add/update/delete actions on database entries4. System updates databases5. Administrator logs out
Post-conditions:	Database is updated
Frequency of usage	Low
Alternative flow of events:	
Exceptions:	
Assumptions:	Administrator has a valid username + password
Issues:	None
Source:	Requirements list
Hardware features:	Webservice for changing data on server

Use Case 4: Update server configurations

Use Case Name:	Updates server configurations
Actor(s):	Service administrator, Server
Description:	Administrator logs in and updates configurations of using Port Authority web-service that allows real-time bus tracking.
Pre-conditions:	None
Normal flow of events:	<ol style="list-style-type: none">1. Administrator logs in to the server.2. Administrator edits configuration entries that defines how the Port Authority services can be reached from the Server3. Server tests configurations.4. Server updates databases.5. Administrator logs out
Post-conditions:	Configurations are updated
Frequency of usage	Low
Alternative flow of events:	Configurations were found invalid while testing at step 3. The system offers to enter different configurations.
Exceptions:	
Assumptions:	Administrator has a valid username + password
Issues:	None
Source:	Requirements list
Hardware features:	Webservice for changing data on server

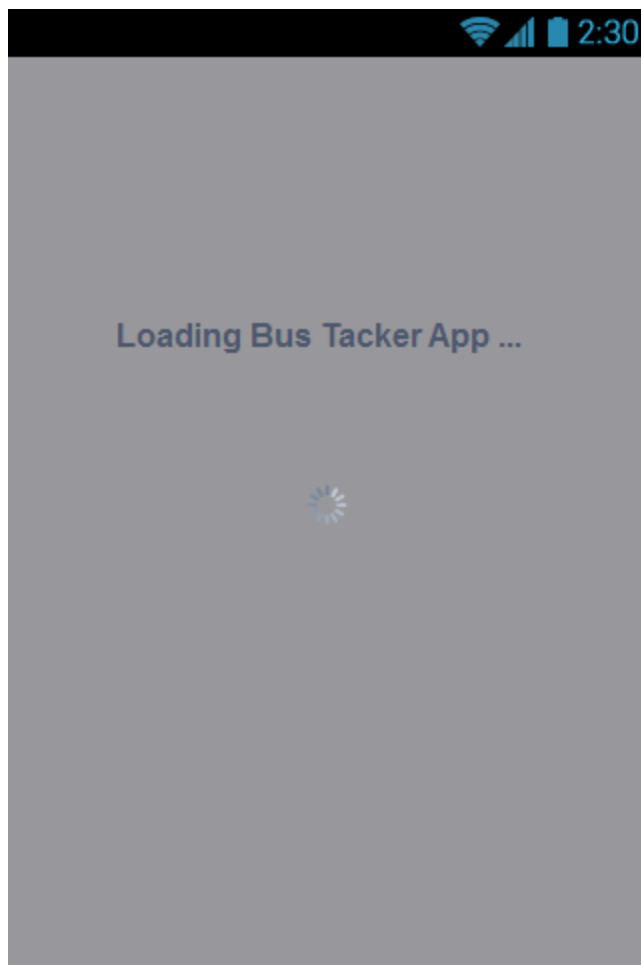
V. UI Illustrations

Loading Screen

The page which is displayed when the app is loading. The app will require time to connect to the server to acquire the current schedule.

Gestures: No gestures or screen input enabled.

Features: Webservice to load bus schedules

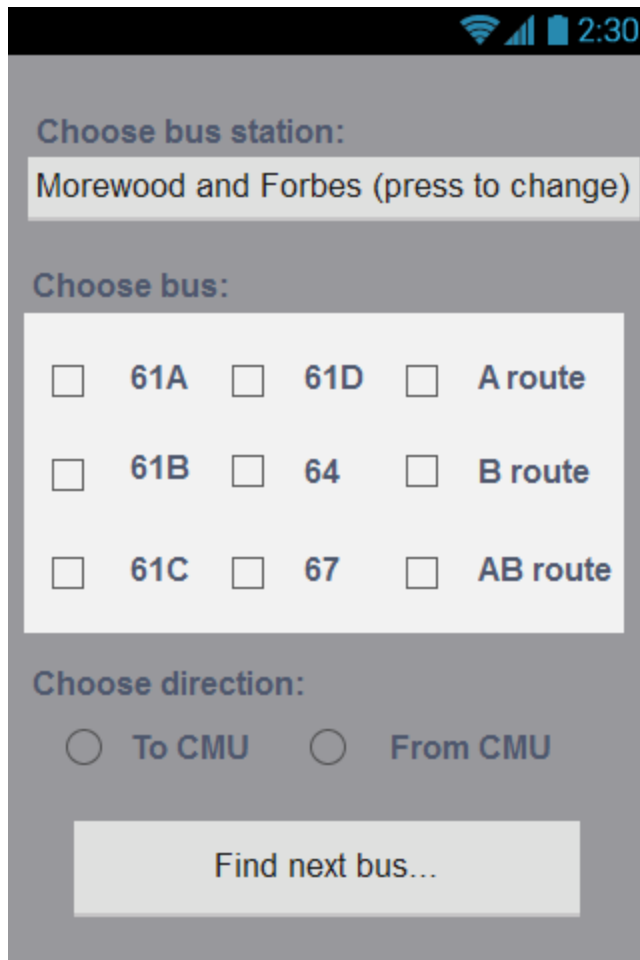


Home Page

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 stop at the station chosen. 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.

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, webservices



The screenshot shows a mobile application interface for selecting a bus. At the top, there is a status bar with a Wi-Fi icon, signal strength bars, a battery icon, and the time 2:30. The main interface has a grey background. The first section is titled "Choose bus station:" in blue text, followed by a white text box containing "Morewood and Forbes (press to change)". The second section is titled "Choose bus:" in blue text, followed by a white box containing a grid of bus options. Each option consists of a small square checkbox, a bus number or route name in blue text, and another small square checkbox. The options are: 61A, 61D, A route, 61B, 64, B route, 61C, 67, and AB route. The third section is titled "Choose direction:" in blue text, followed by two radio button options: "To CMU" and "From CMU". At the bottom, there is a large white button with the text "Find next bus..." in blue.

Choose bus station:
Morewood and Forbes (press to change)

Choose bus:

<input type="checkbox"/>	61A	<input type="checkbox"/>	61D	<input type="checkbox"/>	A route
<input type="checkbox"/>	61B	<input type="checkbox"/>	64	<input type="checkbox"/>	B route
<input type="checkbox"/>	61C	<input type="checkbox"/>	67	<input type="checkbox"/>	AB route

Choose direction:
☐ To CMU ☐ From CMU

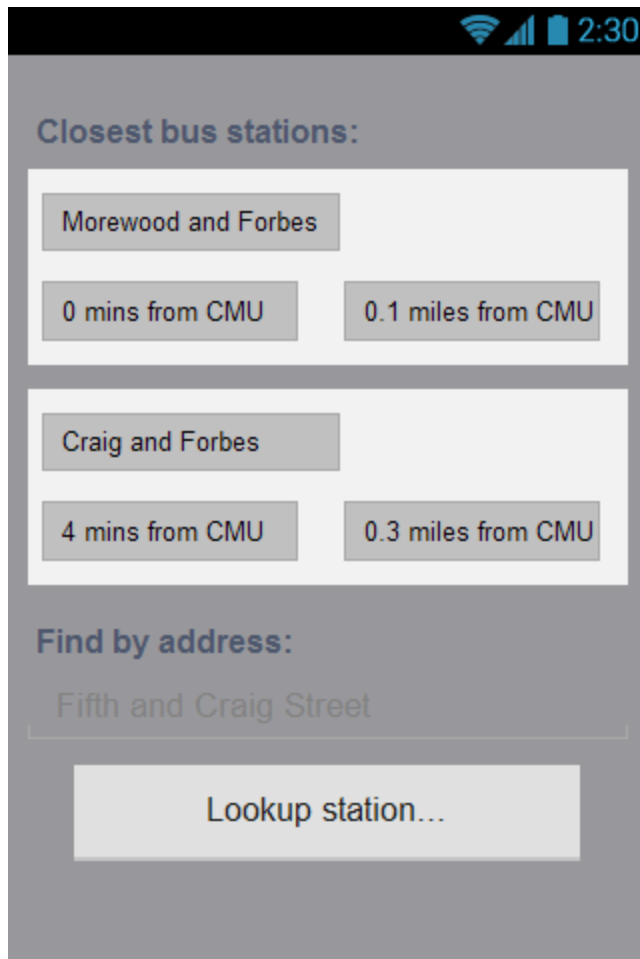
Find next bus...

Locate Stop Screen

This page lists 2 of the closest bus stops. 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.

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, webservices

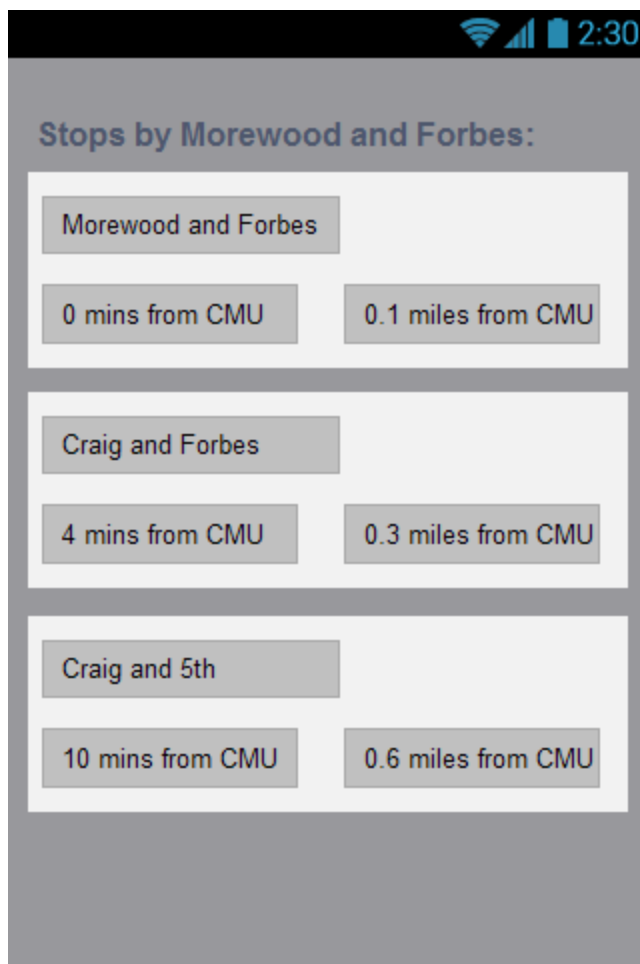


Lookup Stop by Address Screen

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.

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

Features: audio for sound feedback, gestures, touchscreen, webservice

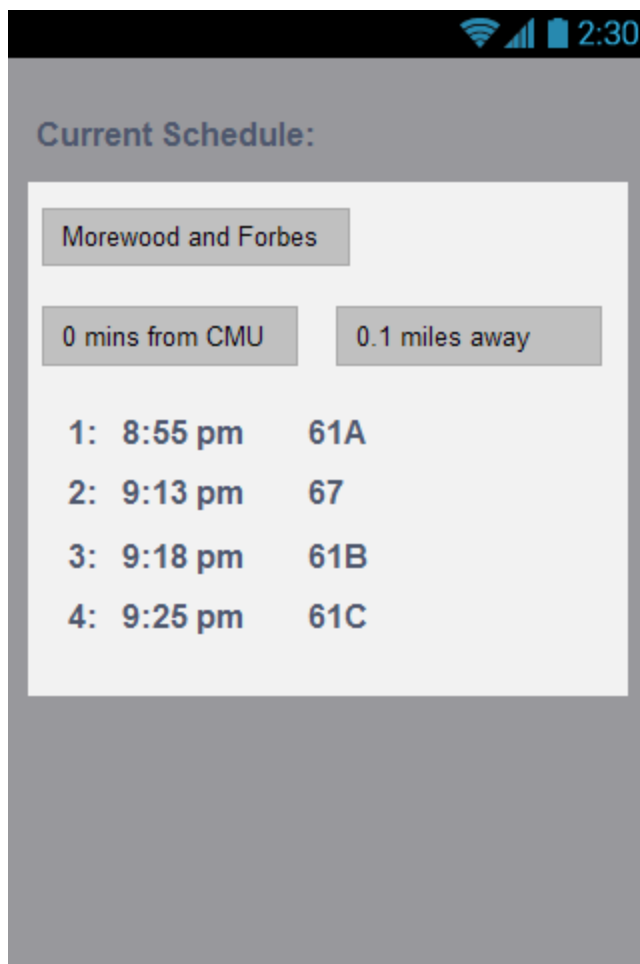


Schedule Screen

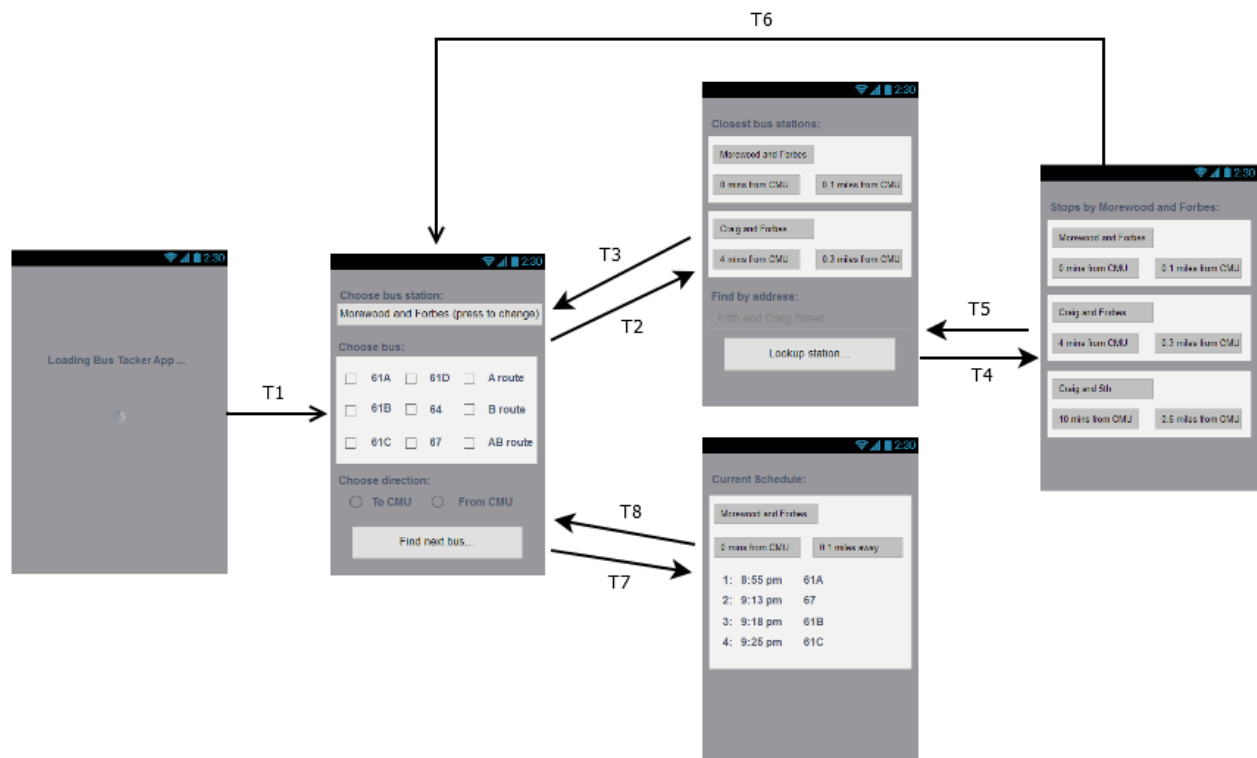
This screen shows the current schedule for the direction and station chosen on the home screen. The list of buses will begin with the next stop of the bus selected on the home screen.

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
T4	User enters or pastes an address into the text field and taps “lookup station” button
T5	User swipes back or presses phone “back” button
T6	User selects a new bus stop from the sorted list of stops

T7	User selects at least one bus and a direction, and taps “find next bus”
T8	User swipes back or presses phone “back” button

VI. References

Tiramisu - Pittsburgh real-time bus tracker project. Based on having people report their GPS coordinates while riding buses:

<http://www.cmu.edu/homepage/computing/2011/summer/wheres-the-bus.shtm>

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>

VII. Business Model Canvas

The Business Model Canvas		Designed for: Bus Tracker - CS Labs	Designed by: Daniel Stoll Ryler Hockenbury Evgeny Toropov	Date: 29/10/2013 Iterations: 1
Key Partners  <small>Who are our Key Partners? Who are our key suppliers? Which Key Resources are we acquiring from partners? Which Key Activities do partners perform?</small> <p>The Pittsburgh Port Authority for maps and schedules of bus routes.</p>	Key Activities  <small>What Key Activities do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue streams?</small> <p>Pulling updated bus schedules from the server</p> <p>Supplying up to date information fro the end users</p>	Value Propositions  <small>What value do we deliver to the customer? Which one of our customer's problems are we helping to solve? What bundles of products and services are we offering to each Customer Segment? Which customer needs are we satisfying?</small> <p>Allows user to save time when traveling by bus</p>	Customer Relationships  <small>What type of relationship does each of our Customer Segments expect us to establish and maintain with them? Which ones have we established? How are they integrated with the rest of our business model? How costly are they?</small> <p>Reliability and endurance by staying up to date with current bus schedules.</p>	Customer Segments  <small>For whom are we creating value? Who are our most important customers?</small> <p>Students and Professors at Carnegie Mellon and University of Pittsburgh</p>
Key Resources  <small>What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue Streams?</small> <p>Up to date information from servers</p> <p>Customers' faith in product</p>	Channels  <small>Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones work best? Which ones are most cost-efficient? How are we integrating them with customer routines?</small> <p>Advertising in app store</p> <p>Information sessions on Carnegie Mellon campus</p>	Cost Structure  <small>What are the most important costs inherent in our business model? Which Key Resources are most expensive? Which Key Activities are most expensive?</small> <p>Marketing, promotion and communications</p> <p>Staff costs</p> <p>Production of application</p>	Revenue Streams  <small>For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How much does each Revenue Stream contribute to overall revenue?</small> <p>Purchases made through app store</p> <p>Expectation from customers will be low pricing</p>	