Project Iteration Overview:

- In this iteration we implemented the following components based on the 3 highest priority big user stories mentioned in iteration 0.
    - Implemented a view for showing near by upcoming busses
    - Implemented a map for showing user's location and near by bus stops
    - Implemented auto information updates based on location change

Git WorkFlow:

- We decided to use the GitHub workflow since it was the simplest most efficient workflow for our type of project. Our project is light on GUI but heavily depends on business logic and services, therefore the GitHub flow seemed like a great fit.
- For this iteration, we created the following branches based on the detailed user stories that we decided to implement
    - Winnipeg-Transit-API (Services)
    - User-Location-GPS (Services/Presentation)
    - Transit-Listview (presentation)
    - Populate-Transit-ListView (business logic)

Group Meeting Logs:

May 19:

- Skype meeting at 3PM
- Duration: 2 hours
- Created Github repo
- Setup access to repo for all members
- Discussed division of responsibilties

May 20:

- Meeting at U of M EITC E2 Windows Lab at 11 AM
- Duration: 3 + ½ hours
- Sketched user interface
- Finalized division of responsibilities
- Began writing developer tasks

May 25:

- Meeting at U of M EITC E2 Windows Lab at 2 PM
- Duration: 3 hours
- Meeting to discuss progress in writing code
- Team members discussed how to use each-others code
- Finalized app name

June 2:

- Meeting at U of M EITC E2 Mac Lab at 4 PM
- Duration: 4 hours
- Finalized ListView row layout design
- Combined ListView and Google Maps View
- Fixed GUI bug that caused crashing when scrolling ListView (array OOB exception)
- Worked on database stub
- Closed git issues
- Did code review

June 5:

- Meeting at U of M EITC E2 lecture room at 12:30 PM
- Duration: 1 hour
- Reviewed each-others progress, tested app on nexus 7
- Began work on readme file
- Worked on testing suite for app
- Decided to move automatic GUI refresh user story to next iteration. However, app will refresh bus schedules when the user changes their location on the map.

Dima Mukhin Individual Log:

May 19:

- Started researching about Winnipeg's Transit API
- Generated an API key
- Mapped all required calls on a paper
- Time spent: 4 hours

May 20:

- Started learning about the Retrofit framework
- Experimented with retrofit on a local test project
- Created a map of dependencies (objects and interfaces)
- Should probably start working on my branch in the next few days
- Time spent: 8 hours

May 21:

- Setup GitHub and cloned project
- Created a branch for the API
- Added API dependencies to the project
- Added Bus-stop related objects
- Implemented API interface
- Time spent 8 hours

May 22:

- Implemented bus-schedule related objects
- Added bus schedule query support
- Time spent: 2 hours

May 23:

- Small changes and fixes
- Renamed TransitClient to TransitAPIClient
- Time spent: 0.5 hours

May 31:

- Code reviews
- Time spent: 0.5 hours

June 3:

- More code reviews…
- Added bus variant support to the API
- Time spent: 2 hours

Abdul-Rasheed Audu Individual Log:

May-21

- Initialized project and created repository
    - Proper naming of packages and files
    - Added necessary files to git ignore to avoid sharing private details
- Time spent: 4 hours

May-22

- Code reviews
- Time spent: 30 minutes

May-25

- Google Maps API
    - Obtained keys and setup maps on screen
    - Added functionality to update location at intervals
    - Clearing and rendering bus stops on the screen as markers
- Time spent: 8 hours

June-2

- Maps and ListView Split Screen
    - Created separate fragments to allow a shared screen for the maps and listview
    - Merged code and resolved merge conflicts
- Time spent: 3 hours

June-4

- Bug fixes
- Merged code from master
- Time spent: 2 hours

Nibras Ohin Individual Log:

May 23:

- Went through the code related to API calls and retrofit framework
- Understood how all the calls were made
- Drew out the hierarchy structure of the objects such as Bus object, Bus routes etc.
- Time spent: 5 hours

May 25:

- Tested out calls to the methods to return nearby bus stops
- Coded method to retrieve detailed information about the buses of the bus stop
- Time spent: 3 hours

May 27:

- Organized received information from the API calls into bus stop name, bus stop number, bus number, bus route and many other components.
- Created and populated a Display object array to feed the information to the listview
- Debugged to test out whether the display object contained the expected information.
- Time spent: 5 hours

May 29:

- Researched on creating a general listview
- Researched on how to have multiple modifiable textviews in one row of a listview for helping out another member
- Followed an online tutorial to make a working example of a listview
- Time spent: 3 hours

June 1:

- Worked with the member creating the listview portion of the app to finalize the outlook of the listview
- Final changes in the listview outlook to accommodate all the information to display in an efficient manner
- Time spent: 4 hours

Paul Artimowich Individual Log:

May 25:

- Installed and updated Android studio
- Got API key from Winnipeg Transit
- Researched how to use Android Studio
- Made github account
- Joined Winnipeg Transit repo app
- Time spent: 2 hours

May 27:

- Read android API documentation
- Time spent: 3 hours

May 28:

- Looked at and ran sample student project
- Read android API documentation
- Read about application life cycle
- Read ListView documentation
- Cloned github Winnipeg transit repo
- Time spent: 4 hours

May 29:

- Discussed with team members where the ListView code should go
- Discussed how it should interact with existing code
- Researched how to construct XML layout files in Android Studio
- Researched how to make a custom row layout for ListViews
- Time spent: 3 hours

May 31:

- Added my Winnipeg transit API key to the code
- Added simple ListView to code
- Created display adapter class
- Tested new code with hard-coded display objects
- Time spent: 5 hours

June 1:

- Modified Display class to give more data that I need for my DisplayAdapter
- Modified DisplayAdapter to extract all needed info from Display class
- Attempted to use ConstraintLayout for ListView rows
- Didn't work because of bugs
- Tried again with LinearLayout, created multiple rows within a ListView row
- All Display information is now displayed within a ListView row
- Fixed bugs
- Time spent: 4 hours

June 2:

- Worked with team members to refine the layout of the ListView rows
- Worked with team members to debug GUI crashes
- (there was an array out of bounds exception)
- Time spent: 3 hours

Syed Habib Individual Log:

May-23

- Code-reviews
- Time spent: 1 hour

May-27

- Business Logic: processing api calls
    - Added functionality to calculate bus remaining time
    - Added functionality to calculate bus status
    - Added functionality to sort the buses
- Time Spent: 5 hours

May-28

- Business Logic: processing api calls
    - Added functionality to calculate bus remaining time
    - Added functionality to calculate bus status
    - Added functionality to sort the buses
- Time Spent: 6 hours

May-29

- Code reviews
- Code reviews fixes for my branch
- Time Spent: 2 hours

May-31

- Integrated presentation (listview) to business logic
- Integrated google maps to business logic
- Time Spent: 5 hours

June-2

- Fixed a bug in listview
- Time spent: 2 hours

June-3

- Created Database Stub
- Time spent: 6 hours

June-4

- Created unit tests for business logic
- Time spent: 9 hours