26 May 2021

Design Specification

Human Decision Making when boarding public transportation

Meher Singh
FINAL YEAR ECSE PROJECT 2021

1.0 Table of Contents

2.0 Document Control

2.1 Revision Control

Version	Date	Details
1.0	23-May-21	Initial version
2.0	25-May-21	Updates based on review with Wynita

2.2 Contributors

Name	Position	Company
Meher Singh	Project Owner	Monash University
Wynita Griggs	Project Supervisor	Monash University

2.3 Approvals

…………………………………………….. Date: …………………………..

Project Supervisor

3.0 Introduction

3.1 Objectives

The purpose of this document is to store and record the design choices made and the ways to achieve the end goals specified in the Requirements Analysis document for the final year project on human decision making when taking public transportation.

3.2 Gantt Chart

The Gantt chart below shows the progression of tasks for the final year project.

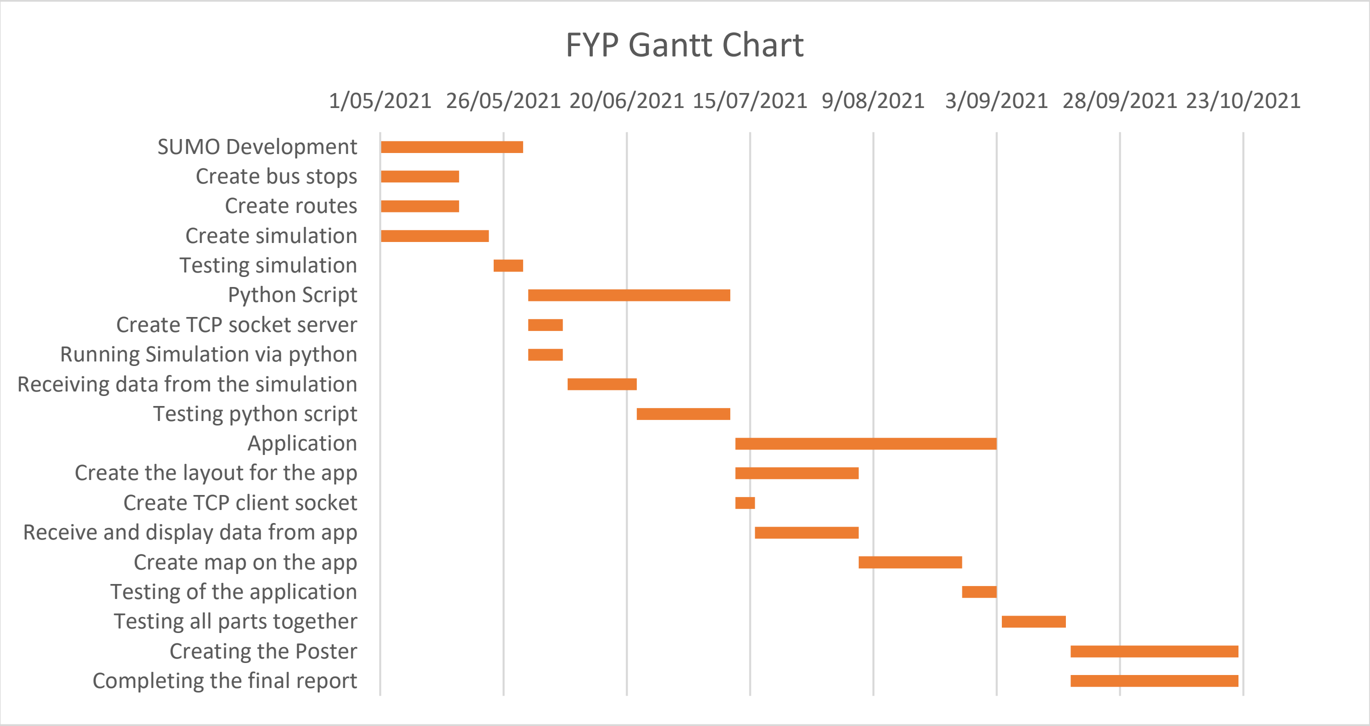

Figure 1: Gantt chart

3.3 Overview of the project

The purpose of the project is to design a tool that can gather data on human decision making regarding public transportation. To achieve this goal, an android application will be made that is able to talk to a server, receive information such as bus times, passengers on-board and seats available, and be able to send data back to the server about their decision on which bus to take as well as a short explanation.

4.0 System Requirements

To run the software's mentioned in 5.0, the following system requirements will need to be met:

OS: Windows 10 64-bit

CPU: x86_64 CPU architecture, 2nd generation Intel Core or newer, or AMD CPU with support for Windows Hypervisor

RAM: 16GB

Storage needed: 16GB

5.0 Software Requirements

The following software’s were chosen to be used for the project:

5.1 Windows

Windows 10 Home

Version: 2004

OS Build: 19041.867

5.2 Android Studio

Android Studio 4.1.2

Build #AI-201.8743.12.41.7042882, built on December 20, 2020

Running on Windows 10

Android SDK: Android 10.0 (Q)

API Level: 29

Revision: 5

Android Virtual Device: Nexus 5x API 29

Compatibility: Windows 8 64-bit or later

5.3 Sumo-gui

Eclipse SUMO sumo-gui

Version: 1.8.0

Compatibility: Windows 10 32-bit or later

5.4 ATOM

Atom text editor

Version: 1.57.0

Compatibility: Windows 7 64-bit or later

5.5 Python

Version: 3.9.5

Compatibility: Windows 8 32-bit or later

6.0 Phone

The device that the app will be running on is a Samsung Galaxy S20 running Android Q (10) OS

6.1 Application Process

The application is being coded in Java using Android Studio. Using TCP sockets, the application will be able to communicate with the Python script running the SUMO simulation. With the socket connection, the app will be able to request information regarding the next arriving busses such as arrival time, number of passengers on board and number of seats available. Once all the data is sent and received by the application, it will allow the user to make an informed decision on whether to take the next bus or wait for another one. The application will then allow the user to send a quick message regarding their decision to the server. Once the server has received the response, it will end the simulation. To provide better performance for the user, the app will show a map of the bus stop and the current locations of the next upcoming busses to that stop. This will be done using OpenStreetMap and the location of the bus from the SUMO simulation.

Initial testing of the application will be completed on a virtual device on android studio before being deployed on the Samsung Galaxy S20.

The following flow chart illustrates how the android application will run and communicate with the python script and SUMO.

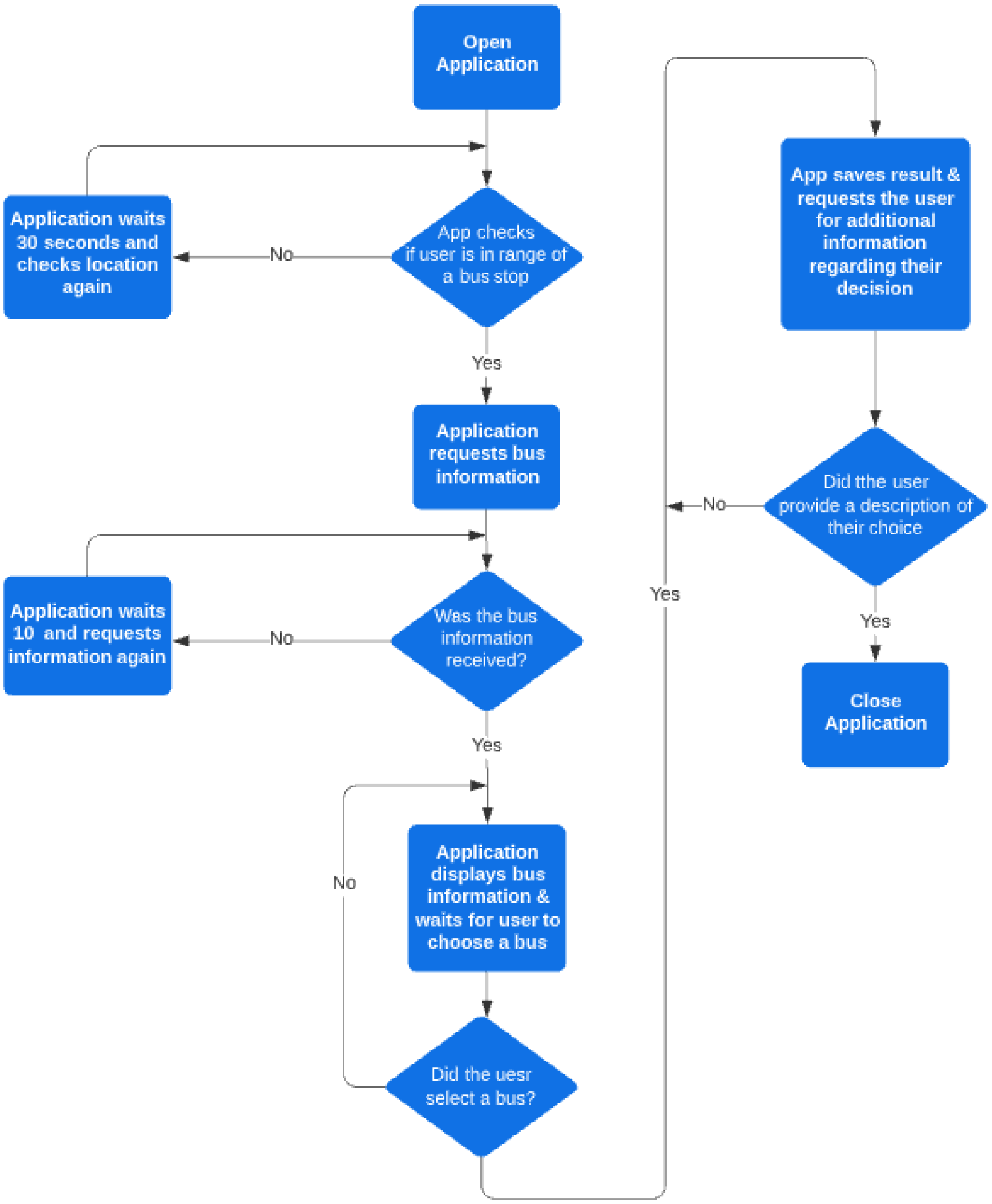
Open Application
Application waits 30 seconds and checks location again
No
App checks if user is in range of a bus stop
Yes
Application requests bus information
Application waits 10 and requests information again
No
Was the bus information received?
Yes
No
Application displays bus information & waits for user to choose a bus
Did the uesr select a bus?
Yes
App saves result & requests the user for additional information regarding their decision
No
Did tthe user provide a description of their choice
Yes
Close Application

Figure 2: Flow chart for the application process

6.2 Sensors

GPS sensors built into the phone will be used by the application to ensure the user is at a bus stop that is pre-defined by the SUMO simulation. This will allow the user to request bus information corresponding to the bus stop they are located in.

7.0 Server/SUMO

The SUMO simulation will be running via a python script that is able to open SUMO and be able to run the simulation for the duration of the testing. The bus route, traffic, and simulated passengers are all pre-defined.

The python script will also host the server that the application is connecting to. Using TCP sockets, the code will create a server socket and connect to the application. Once connection is made, the script will get information regarding busses and the bus stop that the mobile device is at, and transit this data to the app. After this, the code will listen for a response from the application regarding the choice made and the reasoning behind the choice. This data is then saved into a database. For proof-of-concept purposes the database will be a simple text file that the script can output to. This can be expanded upon if required in the future.

The following flow chart illustrates how the python script will run SUMO and communicate with the android application.

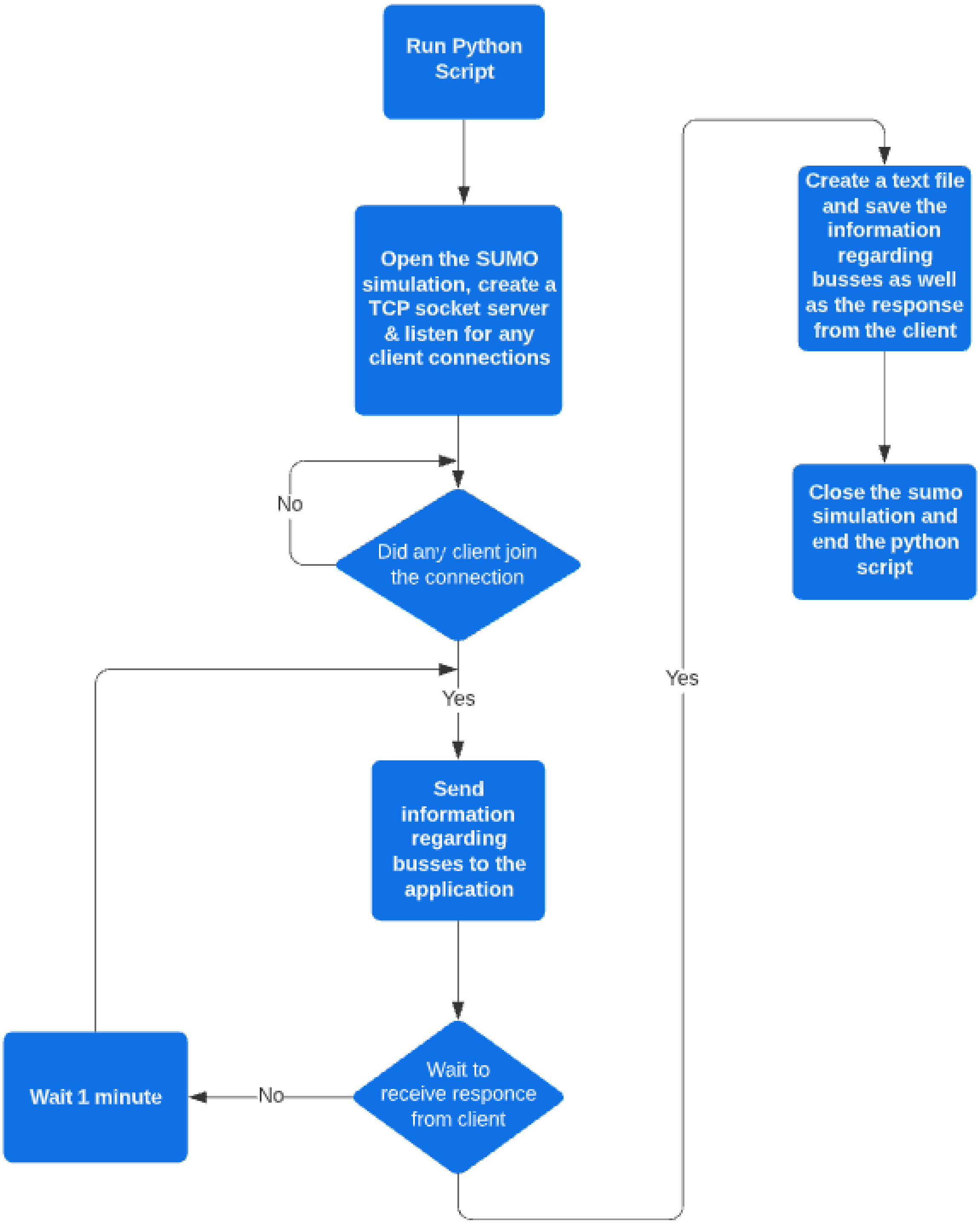
Run Python Script
Open the SUMO simulation, create a TCP socket server & listen for any client connections
No
Did any client join the connection
Yes
Send information regarding busses to the application
Wait to receive responce from client
No
Wait 1 minute
Yes
Create a text file and save the information regarding busses as well as the response from the client
Close the sumo simulation and end the python script

Figure 3: Flow chart for the python script process

8.0 Version Control

A GitHub repository is being used to save and store all files for the project. This repository contains version control so previous changes can be viewed and restored if a new update contains serious bugs.

9.0 Testing

9.1 Application Testing

The testing of the application will be completed on the android emulator built into the android studio software. The emulator will be able to show the app in real time, be able to receive and send information from the server and be able to provide decision making information to the server. Once testing from the emulator is completed, the next stage will be to test the app with the Samsung Galaxy S20 in real world conditions. This involves using mobile data to see how the application functions and to see if communication between the application and the server is consistent.

9.2 SUMO Testing

Testing of the SUMO simulation and python script will be completed on the workstation to ensure that all bugs are found and removed.

10.0 Costs

Costs for this project will involve buying a mobile data plan to prove the application is able to work in the real world without fail. As a phone is already available to use, there will be no costs involved to get the phone.