Department of Computer Science, Faculty of Engineering,
The University of Hong Kong

FYP 16008

A Navigation System for Wheelchair Users

Individual Final Report of Chau Shing Yi

Supervisor
Dr. T.W. Chim

Student
Chau Shing Yi, Sabrina (2013712810)

Another Group Member
Nip Chi Fung, Terry (2013571644)

Date of Submission
16th April 2017
Abstract
Path exploring is not an easy task for wheelchair users. In Hong Kong, many wheelchair users rely on a physical map or Google Maps for navigation. However, those maps may not suitable for this group of users as they have special needs to take care of; Wheelchairs users sometimes suffer from taking the wrong paths because their applications cannot reveal the ‘barriers’ such as staircases in the routes for them.

The objective of the project is going to develop a navigation application *WheelGO* which addresses the navigation issues and provides optimal paths to wheelchair users in a smartphone-based approach.

Like other map applications, *WheelGO* displays turn-by-turn routing instructions that suitable for wheelchair users, indicates accessibility information of each facilities and allows users to report bugs via the feedback form. Besides, the application also has add-on features such as voice guidance function to ensure wheelchair users have a good routing experience.

The paper first describes the necessary background information of the project *WheelGO*, reviews the existing applications on the market and evaluates their strengths and weaknesses in the purpose of enhancing the performance and usability of our application. Then, the methodology and approach of the project, deliverables and difficulties encountered as well as the ideas for further development are going to be discussed subsequently.

Meanwhile, a product with workable functions is ready to be released. With the release of this application, the team hopes to enhance the wheelchair users’ travel experience and to encourage them to explore the outside world.
Acknowledgement

I would like to express our gratitude to my teammate Terry Nip for his hard work and the friendly support provided by our final year project supervisor Dr. T. W. Chim, English class instructor Mr. Ken Ho and the Department of Computer Science, Faculty of Engineering, The University of Hong Kong during the preparation of this final year project.
Table of Contents

LIST OF FIGURES ................................................................................................................. 5
LIST OF TABLES .................................................................................................................. 5
ABBREVIATIONS ................................................................................................................ 5

SECTION I – PROJECT OVERVIEW .................................................................................. 6
  1.1 INTRODUCTION ........................................................................................................ 6
  1.2 EXISTING SOLUTIONS ............................................................................................. 7
  1.3 OBJECTIVES ............................................................................................................ 9
  1.4 TECHNICAL OBJECTIVES ...................................................................................... 10
  1.5 SCOPE ..................................................................................................................... 10
  1.6 TEAM MEMBERS AND STAKEHOLDER ................................................................... 11
  1.7 DEVELOPMENT COMPONENTS .............................................................................. 12
    1.7.1 Equipment Setup ............................................................................................... 12
    1.7.2 Development Tools .......................................................................................... 12
  1.8 TASK LIST, DIVISION OF WORK AND MILESTONES ........................................... 13

SECTION II – APPROACH AND METHODOLOGY ......................................................... 14
  2.1 SOFTWARE DEVELOPMENT CYCLE ...................................................................... 14
    2.1.1 Design Phase .................................................................................................... 14
    2.1.2 Implementation Phase ..................................................................................... 14
    2.1.3 Testing Phase .................................................................................................. 14
  2.2 DESIGN .................................................................................................................... 15
    2.2.1 Infrastructure Overview .................................................................................. 15
    2.2.2 Database Design .............................................................................................. 15
    2.2.3 User Interface Design ...................................................................................... 16
    2.4 Routing Algorithm Design .................................................................................. 16
    2.4.1 Challenge and mitigation ............................................................................... 22
    2.4.2 Developing the basic routing algorithm ......................................................... 22
    2.4.3 Existing problems ............................................................................................ 24
  2.5 Vocal Guidance Design ........................................................................................... 16
    2.6 Test Data and Test Case Design .......................................................................... 17

SECTION III – MAJOR FUNCTIONS SUMMARY .............................................................. 18
  3.1 RECOMMENDS ROUTES FOR WHEELCHAIR USERS (LIMITED TO HONG KONG AREA) 18
  3.2 DISPLAYS TURN-BY-TURN ROUTING INSTRUCTIONS ........................................... 18
  3.3 INDICATES THE ACCESSIBILITY INFORMATION OF EACH POINT OF INTEREST ........ 18
  3.4 PROVIDES VOICE GUIDANCE FUNCTION .............................................................. 18
  3.5 ALLOWS USERS TO REPORT BUGS VIA FEEDBACK FORM ..................................... 19

SECTION IV – CONTRIBUTIONS AND PROJECT DELIVERABLES OF CHAU SHING YI ...... 20
  4.1 SERVER AND DATABASE SET-UP ............................................................................ 20
    4.1.1 Initialize Database Tables ............................................................................... 20
    4.1.2 Build Web Services using Node.js ................................................................. 21
  4.2 ROUTING ................................................................................................................ 22
    4.2.1 Developing the basic routing algorithm ......................................................... 22
    4.2.2 Challenge and mitigation ............................................................................... 22
  4.3 TURN-BY-TURN INSTRUCTION ............................................................................. 24
    4.3.1 Developing the turn-by-turn routing algorithm ............................................. 24
  4.4 AUTOCOMPLETE GEOCODING FEATURE ............................................................. 26
    4.4.1 Limitation ....................................................................................................... 26
  4.5 UI & UNIVERSAL DESIGN .................................................................................... 26
    4.5.1 Difficulty ......................................................................................................... 27
  4.6 VOICE GUIDANCE ................................................................................................... 27
  4.7 ACCESSIBILITY INFORMATION .......................................................................... 28
  4.7.1 Possible Development ...................................................................................... 28
  4.8 DISPLAY USER LOCATION ..................................................................................... 28
  4.9 FEEDBACK ............................................................................................................... 29
  4.10 TESTING .............................................................................................................. 29

SECTION V – PROJECT DIFFICULTIES AND LIMITATIONS .......................................... 29
  5.1 NOT ENOUGH WELL-DEVELOPED OPEN DATA .................................................... 29

SECTION VI – CONCLUSION AND FUTURE WORKS ................................................... 30

REFERENCES .................................................................................................................... 31

APPENDIX I – LIST OF TESTS ......................................................................................... 32
APPENDIX II – SCREENSHOTS OF WHEELGO ............................................................... 33
APPENDIX III – LIST OF DATABASE TABLES ................................................................. 36
List of Figures
Figure 1 A screenshot of the route suggested by the Hong Kong Society for Rehabilitation website ........................................... 7
Figure 2 A screenshot of the Android application 'Wheelmap' showing accessibility information .................................................. 7
Figure 3 Two paths from Knowles Building, HKU to Meng Wah Complex, HKU suggested by Google Maps, both paths include stairs.... 8
Figure 4 Prototype demonstrating the features of WheelGO .................................................. 9
Figure 5-1 Logo of Android Studio .................................................................................. 12
Figure 5-2 Logo of PostgreSQL ....................................................................................... 12
Figure 5-3 Logo of QGIS ................................................................................................. 12
Figure 5-4 Logo of Mapbox ............................................................................................. 12
Figure 6 The infrastructure showing the interactions between the WheelGO and different server of the project ............................ 15
Figure 7-1 The ordinal OSM Map layer (Left) ................................................................... 20
Figure 7-2 The map table entries are rendered on top of the OSM Map layer (Right) ........... 20
Figure 8 List of requests WheelGO can pass to the web service ........................................ 21
Figure 9 Graph with each vertex represents a geographical location ................................ 22
Figure 10 Illustration of the interaction between mobile applications and the Mapbox web server ................................................. 23
Figure 11 Bearing is the angle between magnetic north and the direction of travel ............. 24
Figure 12 Demonstration of three continuous steps in a route ........................................... 24
Figure 13 Icon of WheelGO ............................................................................................. 26
Figure 14 Logo of WheelGO ........................................................................................... 26
Figure 15 One of the screens of the first time tutorial ....................................................... 27
Figure 16 Different dimension for device with different screen size and density ............... 27
Figure 17 The tags imported to the 4 database tables start with 'planet_osm_' ....................... 28
Figure 18 A part of the database table 'feedback' ................................................................. 29
Figure 19 Autocomplete geocoding feature .................................................................... 33
Figure 20 Three stages of showing detailed routing instruction ....................................... 33
Figure 21 Text-to-speech voice guidance and a text pop-up will appear when user click on any mid-point ............................................. 34
Figure 22-1 Red, yellow and green dots represents the accessibility of that location ............ 34
Figure 22-2 Different types of accessibility facilities can be shown independently .......... 34
Figure 22-3 Click event on the coloured dots triggers a text pop-up telling the user the place name and facility type in
(name:type) format .................................................. 34
Figure 23 Show the user location after clicking the 'find my location' button ....................... 35
Figure 24 A text reminder to remind user to turn on GPS in order to use the find my location function ................................................... 35
Figure 25 The feedback form for user to fill in any feedback and send to the developer team ............................................................. 35
Figure 26 The list of all tables in PostgreSQL ................................................................... 36
Figure 27 Table "feedback" ............................................................................................. 36
Figure 28 Table "mtr_exit" ............................................................................................... 36
Figure 29 Table "osm_nodes" .......................................................................................... 36
Figure 30 Table "osm_relations" ...................................................................................... 36
Figure 31 Table "osm_way_classes" ............................................................................... 37
Figure 32 Table "osm_way_types" ................................................................................... 37
Figure 33 Table "planet_osm_line" ............................................................................... 37
Figure 34 Table "planet_osm_point" ............................................................................... 37
Figure 35 Table "relations_ways" .................................................................................... 38
Figure 36 Table "planet_osm_polygon" .......................................................................... 38
Figure 37 Table "planet_osm_roads" ............................................................................... 38
Figure 38 Table "route_bus" ........................................................................................... 38
Figure 39 Table "route_stop" ........................................................................................ 39
Figure 40 Table "spatial_ref_sys" .................................................................................... 39
Figure 41 Table "stop_bus" ............................................................................................ 39
Figure 42 Table "ways" ................................................................................................ 39
Figure 43 Table "ways_vertices_pgr" ............................................................................. 40

List of Tables
Table 1 Comparison of different map products on the market ........................................... 8
Table 2 Smartphone operating system market share report in 2016Q2 ................................ 10

Abbreviations
API Application Programming Interface
HKSR Access guide by the Hong Kong Society for Rehabilitation
OSM OpenStreetMap
Section I – Project Overview

1.1 Introduction

Route planning is a crucial task for navigation. For wheelchair users, the case is even more challenging as there are several unique constraints imposed to the wheelchair users. For instance, in the course of planning, the route planner must not consider any route that contains staircases as most of the wheelchairs cannot climb the staircases. The routes also need to take into account how far the users are able to travel with their wheelchairs and how long is the travel time when exceeded (threshold), wheelchair users may want to travel with public transport instead because a lengthy travel may exhaust some of the users.

Apart from route planning, when traveling to an unfamiliar place, many wheelchair users are also interested in knowing the accessibility information of the facilities. For example, for an underground plaza, people may want to know are there any accessible toilets and are the entrances have only steps or elevators for the incomer to get in.

Through extensive research, it is appeared that there does not exist any effective tool on the market for wheelchair navigation. Many wheelchair users are currently using the existing map products on the market such as the paper map and Google Maps for navigation. However, map products as such do not provide any accessibility information and cannot detect and eliminate staircases in their routes. Moreover, most of the tools require users to regularly stop at every junction to check the maps for direction. Resulting in safety issue and might even make the users fall into a danger situation. Besides, there are some organizations do offer routing and accessibility information look-up services to wheelchair users. For instance, the access guide of the Hong Kong Society for Rehabilitation provides routes planning service specifically to wheelchair users. Wheelmap is a free map that indicates wheelchair accessible places (detailed discussion of the existing products can be found in section 1.2). However, these applications have their limitations and, most of the time, cannot satisfy users’ needs. Many wheelchair users still hesitate to visit an unfamiliar place. In a nutshell, there is a gap in the market for products that can integrate the accessibility information look-up function, and at the same time allow routing planning targeting wheelchair users.

In the following part, the paper will first discuss the other existing solutions and highlight their strengths and weaknesses. Next, the objectives and scope of the application will be illustrated. After that, methodology, brief functions list, detailed work done with the difficulties encountered will be listed out. Based on the work done and the problems encountered during the project development period, the possible future project development will be delivered.
1.2 Existing Solutions

There are several map providers in Hong Kong. The following briefly discusses three applications mainly used by the wheelchair users, namely, Access guide by the Hong Kong Society for Rehabilitation, Wheelmap and Google Maps:

1. **Access guide by the Hong Kong Society for Rehabilitation (HKSR) [4]**

The organization website provides an access guide for wheelchair users to a list of public places (see Figure 1 for the screenshot). Some of the popular attractions have YouTube videos to demonstrate how to get across the roads and buildings. However, the destinations available are limited and the starting points are fixed at public transport stations. Users are not able to choose freely any other starting points and endpoints. This induces to a problem that once the user’s route deviate from the suggested path, he can hardly find his way based on his current location. Moreover, the website does not provide mobile version, it is inconvenience for users to look up the route when he is traveling. This website is not necessarily helpful for the local citizens but nonetheless a great guide in the point of view of a tourist. While different from this access guide, *WheelGo* allows users to dynamically select any two points on the map as the source/destination pair and support Android application, thus foster flexibility.

2. **Wheelmap [1]**

Wheelmap is a free map that indicates wheelchair accessible places all over the world. The map model is based on OpenStreetMap(OSM), a crowdsourcing map project that allows users to contribute by editing the data. It provides valuable information such as the accessibility of each facilities and their contact numbers. Wheelmap does not provide routing service, wheelchair users cannot use the application to obtain suitable routes and need to find the alternatives for routing. In *WheelGo*, not only the accessibility information, but also the feasible routes to destinations will be displayed. Since all the functions are integrated, wheelchair users do not need to check information from different websites.

Google Maps is the most popular navigation application in Hong Kong. The functionalities include turn-by-turn navigation, street view seeing, distances and traveling time measurement between two points etc. The application supports turn-by-turn routing for people with different modes. However, it does not provide routing solution designed for the wheelchair users. The most similar alternative for wheelchair users is the ‘walking’ mode. While wheelchair cannot take full advantage of it because in some situations, all the routes recommended includes waypoints that are not suitable for them. Figure 3 shows the two paths suggested by Google Maps. No route is suitable for wheelchair users as both include staircases. Unlike Google Maps, WheelGO is only designed exclusively for the wheelchair users. Each route WheelGO recommends is suitable for them.

To conclude, Table 1 shows the comparison of strengths and weaknesses among different products on the market, WheelGO summarizes the strengths of each existing product and ready to provide the best service to the users.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>WheelGO</th>
<th>HKSR</th>
<th>Google Maps</th>
<th>Wheelmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Version</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Route for Wheelchair</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Turn-by-turn routing instruction</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>GPS Application</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Voice Guidance</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Accessibility Information</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Table 1 Comparison of different map products on the market

Figure 3 Two paths from Knowles Building, HKU to Meng Wah Complex, HKU suggested by Google Maps, both paths include staircases
1.3 Objectives

The project *WheelGO* is a smartphone-based approach that aims to address the navigation issues encountered by the wheelchair users and to build a free, open and tailor-made navigation solution for wheelchair users.

The preliminary goal is to develop an Android application which can process user requests, acquire accessibility information and suggest barrier-free paths to users (See illustration in Figure 4). With a mobile application, users can enjoy the mobility provided and thus enjoy their journeys more. The application will also provide routing information including time and transportation details. To assist wheelchair users to find path when they are in an unfamiliar place, special facilities such as accessible toilets and ramps, or streets that are inaccessible will be highlighted on the map.

The application is primarily designed for local use only. Accordingly, the application can only process the map requests for the Hong Kong area.

Increase travel satisfaction of the wheelchair users is also one of the objectives of this project. As the application is intentionally build for the wheelchairs, functions and information which are uniquely helpful to wheelchair users are deployed in our application. Voice guidance, and feedback mechanism are therefore implemented. Voice guidance is similar to the GPS driving audio routing, the system will voice out the route according to the users’ real-time GPS location. For feedback system, users can report staircases or places that are not readily accessible for wheelchair users, after receiving the responses, the system then will undergo validation process and update the map so that the accuracy of the routing function can be improved. By adding more functions and features into the application, it is anticipated that the safety and travel satisfaction of the wheelchair users will greatly increase with the aid of one single application *WheelGO*. 

![Figure 4 Prototype demonstrating the features of WheelGO](image-url)
1.4 Technical Objectives

The project also defines some technical objectives with respect to accuracy. The final deliverable should be capable of achieving:

1. **80% Staircase coverage in Hong Kong**
   
   A dataset (database table) is built for storing the locations of staircases in Hong Kong. The limitation of revealing all staircases (excluding indoor corridors) accounts for the application covering only 80% of the staircases in Hong Kong. As agreed by supervisor, brute-force way of achieving map data (i.e. site visit) is inefficient and therefore this application collect staircase data mainly from existing resources on the Internet. Site visit for few districts will be conducted to estimate and validate the staircase coverage.

2. **100% Routing accuracy without staircases**

   Routes computed by the application should be able to avoid staircases in the dataset.

1.5 Scope

1. **The target audience is the general wheelchair users**

   There are wide variety of wheelchair types available on the market, each wheelchair can be categorized according to sizes, mechanisms of control, propulsion method, and technology used for different users’ needs. [10] *WheelGO* can be used by all wheelchair users having any types of wheelchair.

2. **The navigation system implements on the Android platform only.**

   There are two reasons for not choosing to build a mobile version and iOS version. First, get start with the most familiar one. Trying to build an application for iOS while simultaneously building for Android device increase the complexity and the time spent on the project. Since the team is familiar with the android development platform, thus building application with Android first is expected to consume less development time in getting familiar with the development platform. Second, Android currently has the largest global platform share, as seen

<table>
<thead>
<tr>
<th>Period</th>
<th>Android</th>
<th>iOS</th>
<th>Windows Phone</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015Q3</td>
<td>84.3%</td>
<td>13.4%</td>
<td>1.6%</td>
<td>0.5%</td>
</tr>
<tr>
<td>2015Q4</td>
<td>79.6%</td>
<td>18.6%</td>
<td>1.2%</td>
<td>0.5%</td>
</tr>
<tr>
<td>2016Q1</td>
<td>83.4%</td>
<td>15.4%</td>
<td>0.8%</td>
<td>0.4%</td>
</tr>
<tr>
<td>2016Q2</td>
<td>87.6%</td>
<td>11.7%</td>
<td>0.4%</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

Source: IDC, Aug 2016

*Table 2 Smartphone operating system market share report in 2016Q2*
in Table 2, the market share of Android is account for 80-90% of the market share and keep increasing, building an Android application can benefit the most population.

3. The routing service will be built based on the map of Hong Kong area only.
The purpose of this application is to serve the location citizens as well as the tourists come to Hong Kong.

4. The navigation system does not cover indoor map routing.
Indoor map routing requires three-dimension indoor data which is not feasible to collect the data. Most of the facilities are in fact private place that it may have a risk that public is forbidden to view the internal structure of the buildings on the map and result in privacy issue. Besides, the GPS technology nowadays is not mature enough to get the accurate location of the users inside a building.

5. The application has 2 main displayed functions: View Accessibility and Routing
- **View Accessibility**: Indicates the accessibility of the places (e.g. restaurants/shopping malls)
- **Routing**: Shows the point-to-point shortest route across different districts and suggests transport means

1.6 Team Members and Stakeholder
The detailed information of the team is shown in the table below.

<table>
<thead>
<tr>
<th>Team Members</th>
<th>Chau Shing Yi, Sabrina Nip Chi Fung, Terry</th>
<th><a href="mailto:sbnesy@hku.hk">sbnesy@hku.hk</a> <a href="mailto:h1357164@hku.hk">h1357164@hku.hk</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisor</td>
<td>Dr. T. W. Chim</td>
<td><a href="mailto:twchim@cs.hku.hk">twchim@cs.hku.hk</a></td>
</tr>
<tr>
<td>Second Examiner</td>
<td>H. Y. Chung</td>
<td><a href="mailto:hychung@cs.hku.hk">hychung@cs.hku.hk</a></td>
</tr>
<tr>
<td>Project Website</td>
<td><a href="http://i.cs.hku.hk/fyp/2016/fyp16008">http://i.cs.hku.hk/fyp/2016/fyp16008</a></td>
<td></td>
</tr>
<tr>
<td>Project Contact Email</td>
<td><a href="mailto:fyp16008@cs.hku.hk">fyp16008@cs.hku.hk</a></td>
<td></td>
</tr>
</tbody>
</table>
1.7 Development Components

1.7.1 Equipment Setup

- Smartphone OS Platform: Android (with GPS receiver installed)
- Smartphone Minimum SDK: Android 4.4 (KitKat)
- Development Platform: Android Studio
- Map data service: OpenStreetMap (OSM)

1.7.2 Development Tools

The tools used are as follow:

- **Android Studio**
  Android studio is the standard development IDE for developing Android application.

- **PostgreSQL (with PostGIS)**
  PostgreSQL is a NoSQL Object-oriented Database. With its extender PostGIS, PostgreSQL is able to support storing geographic objects and allow queries on the objects. In PostgreSQL Map Data are stored as Points/ Polylines/ Polygons.

- **QGIS**
  QGIS is a geographic information system that allows creating, editing, visualising, and analysing geospatial information.

- **MapBox**
  MapBox is a mapping platform that allows users to visualising map data and do analysis in real time. It provides API (Application Programming Interface) for developers to design their own map in Mapbox Studio.
### 1.8 Task List, Division of Work and Milestones

<table>
<thead>
<tr>
<th>Time</th>
<th>Task</th>
<th>Division of Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>September</td>
<td>Researched on the existing products on the market</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td></td>
<td>Investigated the feasibility of implementation different functions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defined the project scope</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Designed the project webpage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prepared the project plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acquire the necessary development tools</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Brainstormed any additional value-added features</td>
<td></td>
</tr>
<tr>
<td>October</td>
<td>Build app with basic routing function</td>
<td>Terry Nip</td>
</tr>
<tr>
<td></td>
<td>Design and input data into the map database</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td></td>
<td>Draft the tentative UI</td>
<td>Sabrina Chau</td>
</tr>
<tr>
<td>November</td>
<td>Import map data into android studio to build the MapView [Milestone 1]</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td>December</td>
<td>Implement the turn-by-turn routing function</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td>January</td>
<td>Continue implementing the turn-by-turn routing function [Milestone 2]</td>
<td>Sabrina Chau</td>
</tr>
<tr>
<td></td>
<td>Write the interim report</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td>February</td>
<td>Implement acoustic navigation function [Milestone 3]</td>
<td>Sabrina Chau</td>
</tr>
<tr>
<td>March</td>
<td>Implement the feedback mechanism [Milestone 4]</td>
<td>Sabrina Chau</td>
</tr>
<tr>
<td></td>
<td>Improve the UI design of the app</td>
<td>Sabrina Chau</td>
</tr>
<tr>
<td></td>
<td>Improve the routing algorithms</td>
<td>Terry Nip</td>
</tr>
<tr>
<td></td>
<td>Test and debug</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td>April</td>
<td>Finalize the implementation [Milestone 5]</td>
<td>Sabrina Chau, Terry Nip</td>
</tr>
<tr>
<td></td>
<td>Write final report</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prepare the exhibition poster</td>
<td></td>
</tr>
</tbody>
</table>
Section II – Approach and Methodology

2.1. Software Development Cycle

The project development follow a software development cycle framework. The cycle is divided into three phases, namely, design phase, implementation phase, and testing phase.

2.1.1 Design Phase

The design phase is already completed in October. At this stage, the project team was focus heavily on studying the feasibility of implementation different functions and devising the application architecture. The details of design phase can be found in Section 2.2.

The system design phase includes the following works:

- Brainstorm any value-added functions that wheelchair users need
- Decide the software used
- Design the routing algorithm
- Design the database
- Design the user interfaces
- Design the testing area

2.1.2 Implementation Phase

The implementation phase is conducted during November to mid-March, where the features and functions designed is implementing into the WheelGO. The details of design phase can be found in Section 3 and 4.

2.1.3 Testing Phase

The testing phase is carried out at late March to mid-April. Testing has undergone a series of tests that test all components of the system as well as test for the precision of the routing algorithm. The details of design phase can be found in Section 4.10.
2.2 Design

2.2.1 Infrastructure Overview

Figure 6 shows the overall infrastructure illustrating the application data flow. When users enter WheelGO, the application send request for the mapView to MapBox using the API provided by MapBox. MapBox is also responsible for handling the autocomplete geocoding function. The team has also run its own web service by self-build web server, when users perform accessibility information request or routing request, WheelGO send a web service request to the web server. After querying the database server and processing, the web server response to WheelGO in JSON format.

2.2.2 Database Design

A NoSQL spatial database PostgreSQL is chosen to be the database for the project. As the map dataset tends to be extremely large, to avoid storing gigabytes of tiles resulting in slow processing rate and difficulty in storage, the design of database needs to consider three factors:

1. data load & preparation for data analysis,
2. data querying and
3. data storage.
And there are three types of data the service needed to process and store:

1. Map data – map view data and transportation data
2. Accessibility information – The information of the accessibility of different facilities such as the accessibility of toilet, some of the data relies on public contribution to update
3. User feedback

For the ordinal data, e.g. the accessibility information and user feedback, almost any database system on the market can be used to store them. However, for the map data, PostgreSQL (with extension PostGIS to provide spatial features) can smartly vectorizing and rendering the map data into the spatial database. It has enormous advantages over the conventional raster-based visualization and can improve the overall performance when handling queries from a mobile application [6]. Therefore, Both the ordinal data and spatial data are stored in different tables in the same database.

2.2.3 User Interface Design

As the targeted users are wheelchair users. The principle focus of the UI design is to create a simple, neat and automatic application. This is because involving less interaction can improve the accessibility for wheelchair users. The approach will be to automate some of the elements during routing, such as automatically track the user’s location during routing and output the notification sound instead of requiring users to interact with the application every time they cross a junction.

2.2.4 Routing Algorithm Design

To start with, the application needs to design a shortest path algorithm that eliminates the unsuitable path. The application will make use of the GPS technology and other A-GPS technology to calculate the shortest candidate paths. The candidate algorithm is A* algorithm [8], a single-source shortest path algorithm that calculates the path in a better worst case performance O(|E|) as compare to O(|E|+|V| log |V|) in Dijkstra algorithm [9]. However, different from the original A* algorithm which only keeps track of the shortest path, the algorithm needs modify for our case to suit the special requirements, such as the new algorithm should validate whether the path contains any prohibited waypoint during traversing the node.

2.2.5 Vocal Guidance Design

The vocal guidance is used to guide the users during navigation. The application will use Google Text-To-Speech technology to generate a synthesized voice that outputs the direction to the user.
2.2.6 Test Data and Test Case Design

The application will be assessed in term of the robustness of the routing algorithm, the functionality of the application as well as the correctness of the dataset at the testing phase.

Since the routing algorithm is to be defined by the team, thus it is needed to design a series of test cases to examine the algorithm. The test will be conducted based on various location in Hong Kong, the application will set random start and end points to check whether the routes are correctly set.

The reasons for conducting the case study against the correctness of dataset are because there does not exist an academic consensus on the standard of how to assess the barrier-free access level of each facility [7]. For this reason, tests are needed to assess whether the paths recommended by the application are feasible. A case study involving around five wheelchair-dependent paraplegics will be held at testing stage to assess whether the recommended paths are appropriate for wheelchair users to use. The detailed test areas and passing criteria can be found in Appendix I.
Section III – Major Functions Summary

3.1 Recommends routes for wheelchair users (limited to Hong Kong area)

*WheelGO* has 3 transportation modes to provide possible shortest route between any two points within Hong Kong, and they are namely wheelchair mode, bus mode and MTR mode.

The wheelchair mode is similar to walk mode of Google Maps while the main difference is with all the staircases removed and the travelling distance is restricted to 1000m. For bus mode, the route provides maximum up to 2 bus transfers and displays the route in the form of wheelchair mode turning points and bus stops. The MTR mode displays the route from source to MTR exit, and the other route from the other MTR exit to the destination. When users want to switch from one mode to another, they can click on the same button and the status exchanges between 🚶‍♂️, 🚌 and 🚇.

3.2 Displays turn-by-turn routing instructions

Users may have trouble while reading route directly from the map. Hence, turn-by-turn instruction is needed for giving a better verbal guidance to the wheelchair users. *WheelGO* provides step-by-step instructions in how to go from point A to point B to guide the way and directions to the users.

3.3 Indicates the accessibility information of each point of interest

*WheelGo* provides accessibility information display. As shown at Figure 21-1. The three icons 🚶‍♂️, 🚴‍♂️ and 🚷 represent not accessible, limited accessibility and accessible for wheelchairs. The information can be further categorised as different categories according to their different facilities types as shown as Figure 21-2.

3.4 Provides voice guidance function

Currently, the text-to-speech voice guidance provides in 2 situations: in turn-by-turn voice guidance and in displaying accessibility information. In turn-by-turn voice guidance, when the users navigate to a place, either by click the mid-points or the turn-by-turn instruction, a text pop-up and a voice direction are generated (See Figure 20). For displaying accessibility information, when the users click on the coloured buttons 🚶‍♂️, 🚴‍♂️ and 🚷, text pop-up and a voice speaks aloud the facility’s name and type (See Figure 21-3).
3.5 Allows users to report bugs via feedback form

To ensure the database is most updated, sometimes we require users to help us to maintain the most accurate data. *WheelGo* is a navigation tool. The database may be outdated and the developer cannot know the full picture. Therefore, a feedback form is required. When the users click on the button, the feedback form is shown for user to input feedback (See Figure 25).
Section IV – Contributions and Project Deliverables of Chau Shing Yi

4.1 Server and Database Set-up

The server is put on the virtual server provided by the CS Department. By installing the PostgreSQL database (with extension PostGIS) and setting up the node.js server, the server provides a web service and listens to port 8080. Upon user perform actions from WheelGO, the event handlers listen to the action performed and send the corresponding post requests to fyp16008s1.cs.hku.hk:8080/[params], after processing, the server will send the corresponding response back as JSON format. The full list of database tables, table attributes and request types can be found at the appendix III.

4.1.1 Initialize Database Tables

The database consists of 17 database tables. Every subset of tables is served as different purposes. Of which, 11 of them are processed by me, tables ‘osm_nodes’, ‘osm_relations’, ‘osm_way_classes’, ‘osm_way_types’, ‘ways’ and ‘ways_vertices_pgr’ are created by using converter ‘osm2pgrouting’ to convert the .osm file grabbed from www.geofabrik.de and served as route calculating propose; tables ‘planet_osm_line’, ‘planet_osm_point’, ‘planet_osm_polygon’ and ‘planet_osm_roads’ are created by using another converter ‘osm2pgsql’ which stored the map component as points, polylines and polygons in different tables. According to the type and specific usage, these tables represent the layers rendered on the map. Figure 7-1 is the ordinal OSM map and Figure 7-2 is the map data stored. Purple area, blue lines and red points are the location geography stored in the tables ‘planet_osm_polygons’, ‘planet_osm_polylines’ and ‘planet_osm_points’ respectively. These tables are useful for displaying accessibility information propose; table ‘feedback’ is used to store user feedback submitted from the feedback form function inside WheelGO.
4.1.2 Build Web Services using Node.js

The server supports 10 kinds of requests (As listed as Figure 8). My part are /access_info, /feedback and part of /nearnode and /route. For /access_info request, server receives user request, query on the tables ‘planet_osm_line’, ‘planet_osm_point’, ‘planet_osm_polygon’ and ‘planet_osm_roads’ of the database, based on the query results, distinguish the facility type and return the name, accessibility information and facility types to the user in JSON format. For /feedback request, server receives user feedback as a Feedback object format. Then, it connects and insert the feedback to the table ‘feedback’ and return response as ‘ok’ or error types to the user in JSON format. For /nearnode request, as the routing algorithm need to base on the graph nodes and edges to calculate the shortest distance, the request need to supply 2 parameters, latitude, longitude the source/ destination, the server pass the query to the database and return with the nearest OSM node ID and its latitude, longitude. For /route request, it is the most substantial part of the project. It consists of 4 parts and need to be processed one-by-one. First, Terry calculated the geometry of the shortest route first. Using the route geometry, I use the geometry to calculate the instructions including the directions and each instruction street name and return the route’s geometry and maneuver (including bearing, location, instruction, direction of each steps) to WheelGO.
4.2 Routing

4.2.1 Developing the basic routing algorithm

The routing function is of the cooperation of the teammates while I mainly do the earlier stage implementation and Terry focused on further developing the bus and MTR route and the refinement and betterment of the algorithms. In this project, we decided to apply Dijkstra’s Algorithm, which requires extra storage of graph component. Figure 9 is a simple graph representing a city map. Each vertex represents a geographical location, while an edge illustrates a path between two locations. By applying Dijkstra’s Algorithm, the shortest path between two locations can be obtained. The major benefit of this approach is that the resultant path can be changed by adjusting the graph component since the graph is defined by the team.

4.2.2 Challenge and mitigation

4.2.2.1 Google Maps APIs are not applicable

The project was planned to co-operate closely with the Google Maps APIs. The map shown in our user-interface was implementing the Google Maps API, while the working principle of our routing function is select and reject routes obtained from Google Map Direction API. However, according to the Google Maps/Google Earth APIs Terms of Service [3], real-time navigation or route guidance implementing these APIs are prohibited. The audio navigation function, one of the key features in the project, is originated from real-time navigation. Consequently, implementing the feature will lead to violation of the Google Maps/Google Earth APIs Terms of Service.

It is possible to overcome the problem by two solutions, and they are: (1) removing the audio navigation feature, and (2) replacing the Google Map APIs by substitutes. Comparing the two approach with respect to the level of difficulty, removing the audio navigation feature is apparently easier as less work is needed to be done. Besides, much time can be saved as finding suitable substitute can be avoided. However, the first approach will sacrifice the overall functionality of the application, and adding this feature will definitely benefit the users. Therefore, other Map APIs were considered for developing our application.
4.2.2.2 Mapbox Direction APIs are not preferable

The immediate substitute for Google Map API was Mapbox API and the approach for obtaining routes became using the Mapbox Directions API. Mapbox is powered by the open data obtained from OSM. Figure 10 illustrates how mobile application obtains route information from the Mapbox web server. After receiving a HTTP request with coordinates, the server with return the routes in JSON format. The returned JSON object contains detailed route information including duration, distance and step.

Mapbox Direction API was implemented. Some tests were conducted and the JSON objects obtained from the Mapbox Direction API were analyzed. The Mapbox Direction API supports routing profile which represents the desired type of transportation. Three profiles (driving, cycling and walking) are available in the Mapbox API and the server will only return paths which are feasible to the particular profile. For instance, the driving profile shows the fastest routes by preferring high-speed roads, like highways. However, there is no profile preferring public transportation such as MTR and bus in this API. In other words, the API do not take into consideration public transportation in Hong Kong when computing paths.

Besides, the number of paths returned per request is limited (A maximum of 3 paths is returned). The team tested average number of paths returned by inputting several source and destination pairs. However in reality, only 1 path was returned for most of the test cases. For instance, no routes were displayed for some test cases as the only returned paths was rejected for the staircase discovery.

In conclusion, Mapbox Direction APIs are not preferable and the team is redesigning the routing algorithm running on top of the map database created by the team using PostgreSQL. By dynamically adjusting the map components inside the map database, e.g. remove all staircases data in the map database, it can be ensured that the routing algorithm can avoid generating results that include any barriers such as staircases and escalators.
4.3 Turn-by-turn instruction

4.3.1 Developing the turn-by-turn routing algorithm

In this project, the turn-by-turn instruction mechanism is self-defined instead of using API provided from third party. The turn-by-turn instruction can be displayed in a bottom sheet and pull out as shown as Figure 20. To implement turn-by-turn instruction function, the instructions should go through pre-processing before outputting to the screen. Currently, the turn-by-turn instruction is being calculated separately in 2 parts.

i.e Turn sharp right on 山道 Hill Road

The first part labelled in yellow colour shows the direction and the latter part labelled in blue colour displays the street name. My job involves calculating the orientation on my own. The algorithm is originated from open-source route planning application Graphhopper’s direction algorithm. The direction is calculated by first calculating the bearing (as shown in Figure 11, the angle between North and the next turn mid-point direction the user facing):

\[
\begin{align*}
\text{var } y &= \text{Math.sin(lon-prevLon)} \times \text{Math.cos(lat)}; \\
\text{var } x &= \text{Math.cos(prevLat)} \times \text{Math.sin(lat)} - \text{Math.sin(prevLat)} \times \text{Math.cos(lat)} \times \\
\text{Math.cos(lon-prevLon)}; \\
\text{bearing} &= \text{Math.round}((\text{Math.atan2}(y, x) \times (180 / \text{Math.PI}));
\end{align*}
\]

Then, to predict the direction for the next step, we need three points to make directions. Namely, next point, previous point and double previous point as shown in Figure 12 which 0 is the node visited at t-2, 1 is the node visited at t-1 and 2 is the node being visited at instant t. By calculating the angle of a turn, defined by the three points.

![Figure 15 Bearing is the angle between magnetic north and the direction of travel](image)

![Figure 16 Demonstration of three continuous steps in a route](image)
The above code fragment is the algorithm we use. orientation is the angle of the vector from 1 to 2 expressed as atan2, while previous orientation is the angle of the vector from 0 to 1. Intuitively, if orientation is smaller than previous orientation, then we have to turn right, while if it is greater we have to turn left. To make this algorithm work, we need to make the comparison by considering orientation belonging to the interval \([-\pi + \text{previous orientation}, +\pi + \text{previous orientation}].

As for the latter part of the instruction, the sign calculated above are then pair up with the street name retrieved from the edge name using the tables ‘osm_nodes’ and ‘ways’.

### 4.3.2 Existing problems

At present, WheelGO has one defect in calculating the turn-by-turn instruction. That is the OSM data does not save any information about roundabout. While without the roundabout information, the paths pass through the roundabout would present a wrong direction due to
the restriction of the algorithm. Not until the OSM data include the roundabout information, the team cannot provide a solution for the problem.

4.4 Autocomplete Geocoding feature
In map, every location is representing as a single point or a list of bounding geographic coordinates (latitude & longitude). Geocoding is the process of converting addresses into geographic coordinates, so that the users can enter an address and the system translates the address to a point on map. In WheelGO, using the Mapbox geocoding API, users can either enter from the autocomplete geocoding text box (See Figure 19) or directly pointing to the map to select source and destination for navigation.

4.4.1 Limitation
The Mapbox database does not have ample amount of Hong Kong geo-location data, that makes search result limited. The current alternative is to directly click on the map for routing.

4.5 UI & Universal design

Icon
The icon of the WheelGO depicts a wheelchair user exerting for a spurt across the white line to the destination. The design idea is on one hand, shows the determination of wheelchair users using every mean to try their best to achieve their goal; on the other hand, reflects the main objective of WheelGO, to assist wheelchair users to arrive their destination.

Logo
The logo of WheelGO having a bright colour theme with a map market replacing the ‘O’ character because wants to give a positive and clear image to the users that the application is used for map navigating.
First-time Tutorial

For easy understanding of the navigation interface. The system provides a first-time tutorial of the interface. Figure 15 shows the screen captured. Users are provided with explanation of the meaning and usage of the textboxes and buttons. The application also contains features take care of user experience.

Size, Order and Show/Hide of the UI Components

The sizes of the UI Components such as the buttons and auto complete text boxes are set to be minimum touchable size to avoid blocking the map view. Besides, the order of the UI components is carefully planned, for example in Figure 15, the mini-sized floating action buttons are arranged according to their possible use rate, such as the ‘transport mode’ button is closest to the ‘toggle open menu’ button. Finally, the most importance function of WheelGO is navigation. Therefore, the auto-complete text bars function is shown while the user enters the application. The UI component will automatically force to hide if the UI component is not in use. For example, After the user has entered both source and destination, the detail instruction bar below will show and the autocomplete bar will be hidden so that the complete routing will not be blocked. The other functions, except for the zoom in/out functions are hidden and fold inside a single button reside at the bottom right corner of the application.

4.5.1 Difficulty

Components’ positions appear differently across different screen size. The reason is because of the density and dimension of devices are different. The margins set for each component vary. The problem solved from reading along the Supporting Multiple Screens. The team have adjusted the dimension for each device density (See Figure 16) to minimize the differences across different devices.

4.6 Voice Guidance

The voice guidance is provided by Google Text-to-speech which is a screen reader that speak out the turn-by-turn instructions during routing mode and accessibility information during view accessibility mode. Currently, WheelGO only provide English mode. At the application starts,
A ‘TextToSpeech’ object is called to get ready for receiving the voice output request. When the user click on the accessibility markers or turn-by-turn stop markers, an event listener is triggered and the words for the markers title will be speak aloud.

4.7 Accessibility Information

The data is come from OpenStreetMap internal data. OSM consists of a long list of identifying called ‘Tag’ inside the .osm file. By using importing tool osm2pgsql, points, polygons and poly lines representation of the map component are stored as table ‘planet_osm_roads’, ‘planet_osm_polygons’, ‘planet_osm_polylines’ and ‘planet_osm_points’ respectively, as well as the specified tags can be imported into the database. For this project, as shown in Figure 17, the name tag, wheelchair related tags and facilities type tags are chosen for processing the accessibility related information. When the user click on the get accessibility info buttons. The event handler send a post request (via the URL fyp16008s1.cs.hku.hk:8080/access_info) to the server and the server then return the place information as shown below:

```json
{"place":[
  {
    "name": "天壇大佛 Tian Tan Buddha Statue", "wheelchair": "no", "lat": 22.253990797058,
    "lon": 113.905045284144, "type": "artwork"},
  {
    "name": "駿運北路 Chun Wan Road North", "wheelchair": "limited", "lat": 22.2950436970532,
    "lon": 113.923366284142, "type": "bus_stop"},
  ...
  (deleted for simplicity)
]}
```

All POIs having accessibility information will have a coloured marker is then being rendered as the respective latitude and longitude on the map (See figure 22-1 in Appendix II).

4.7.1 Possible Development

The marker display can be further refined to show the categories from the map. Besides, the markers may allow users to edit the accessibility information directly on the map instead of the current approach: to reflect the willingness of change by filling the feedback form.

4.8 Display User Location

Users can toggle their current location on the map using the button depending on the state of the button. The function is achieved by applying Mapbox ‘LocationServices’ object to check whether the user location permission has been granted. And then call a location listener to move the camera to display user location on the map by setMyLocationEnabled().
4.9 Feedback

The feedback form is a DialogFragment as a form of pop-up window (See Figure 25 at Appendix III). When users want to give feedback to us, they can click on button and input the contact information, select response problem type (‘Accessibility Information’, ‘Route’, ‘Suggestion’ or ‘Others’) and write any response to us. The form will be encapsulated in a post request (via the URL fyp16008s1.cs.hku.hk:8080/feedback) and send to the server (Figure 18 demonstrates a part of feedback the server received). Upon receiving input, the server send back the status response to the user. After that, the develop team will check validity of the barrier report, and do manipulation on the existing dataset subsequently.

![Table]

*Figure 22 A part of the database table 'feedback'*

4.10 Testing

Android runs on a variety of devices with different screen sizes, densities and performance. Therefore, apart from testing for general bugs of the application, the team also focus on ensuring the consistent User Interfaces appear on different devices by testing and adjusting the dimensions with respect to different devices. The team have conduct a list of tests as stated at Appendix II.

Section V – Project Difficulties and Limitations

5.1 Not enough Well-Developed Open Data

Although Hong Kong Government is promoting data open access, different government departments have dedicated spatial data, for instance, the GeoInfo Map provided by Lands Department. The data support is still insufficient. For example, the MTR lacks the exit data. Furthermore, open data by provided by the government is not developer-friendly. Although the eTransport, which developed by Transport Department and provides point to point public transport route enquiry, it does not provide API for open-access of the data for software developers. Most of the public information are presented in Excel, CSV or PDF format and does not provide API. These reasons forms obstacle for public usage.
Section VI – Conclusion and Future Works

This report presented the design motivation, requirements and approaches of building a navigation application dedicated to wheelchair users. The application is different from the other products on the market as it supports real-time voice-guided routing and eliminates the paths that wheelchair users might not able to follow such as the paths that contain staircase. To effective maintain and expand the database of the locations that wheelchair users are not able to pass through, the application also offers a feedback system. By crowdsourcing the data, the database can be completed quickly and hence all other users may benefit from it.

After extensive study on the previous works, it is anticipated that the implementation should be highly feasible and the application will be considerably useful to the wheelchair users. The reason is because the application, if successfully implement the vocal guidance feature, can effectively lower the frequency of them to pause and check the map, hence lower the accident rate.

Furthermore, although there might have been issues on the not enough data. Our team has concluded the most important deliverable for wheelchair users is the ability of the application to recommend a feasible route for them.

Although the proposed objectives on the WheelGO project have been achieved, there is plenty of room for improvement. The future development of the WheelGO project should focus on 2 aspects. First, develop the application in iOS version. Android account for the largest market shares of the mobile phone market, while there are also a large group of people using iOS mobile system. Therefore, building both iOS and android versions should serve all the wheelchair users’ needs. Second, improve the quality of the routes suggested. In order to improve the quality of the routes suggested, more studies on the design of routing algorithm are need.
References


Appendix I – List of Tests

The table below shows the test areas as well as the passing criteria the testing phase will test against:

<table>
<thead>
<tr>
<th>Test area</th>
<th>Description</th>
<th>Passing criteria</th>
</tr>
</thead>
</table>
| Routing Accuracy                 | Test whether the route knows to take public transportation when users input a cross-district source/destination pair. | 1. no staircases shown  
2. the route takes public transportation instead of walking  
3. Pass 30 test cases of random source/destination pair                                                                 |
| Routing Accuracy                 | Test whether the route knows to take public transportation when users input a source/destination pair with travel time > 20 minutes. | 1. no staircases shown  
2. the route takes public transportation if any  
3. Pass 30 test cases of random source/destination pair                                                                 |
| Feedback                         | Test whether the users can successfully send feedback to the server.         | 1. the users can successfully send feedback to the server.  
2. Pass 5 test cases from 5 different Android phones                                                                 |
| Turn-by-turn routing             | Test whether the turn-by-turn routing instructions can be correctly displayed. | 1. The instructions are properly shown on the screen  
2. The turn-by-turn paths are feasible for wheelchair users to walk through  
3. Pass 30 test cases of random source/destination pair                                                                 |
| Voice guidance                   | Test whether the voice guidance system pronoun all the words appears on the turn-by-turn instructions. | 1. All the words on the routing instructions (including the address name) can be pronounced  
2. Pass 30 test cases of random source/destination pair                                                                 |
| Voice guidance                   | Test whether the voice instructions can switch automatically after it detected the position changes of users. | 1. The delay of the voice instructions is not more than ±10 seconds  
2. Pass 30 test cases of random source/destination pair                                                                 |
| Accessibility information        | Test whether the accessibility information is located at the correct corresponding locations. | 1. The accessibility information of each place is properly located  
2. Pass 5 test cases of random source/destination pair                                                                 |

Table 4 Test areas and passing criteria for the testing phase
Appendix II – Screenshots of WheelGO

Autocomplete Function

![Autocomplete geocoding feature](image)

Figure 23 Autocomplete geocoding feature

Turn-by-turn Routing Instruction

![Three stages of showing detailed routing instruction](image)

Figure 24 Three stages of showing detailed routing instruction
Voice Guidance Function

Figure 25 Text-to-speech voice guidance and a text pop-up will appear when user click on any mid-point

Accessibility Information Function

Figure 28-1 Red, yellow and green dots represents the accessibility of that location

Figure 27 Different types of accessibility facilities can be shown independently

Figure 26-3 Click event on the coloured dots triggers a text pop-up telling the user the place name and facility type in name\:type format
Find My Location Function

Figure 30. Show the user location after clicking the 'find my location' button.

Feedback Function

Figure 29. A text reminder to remind user to turn on GPS in order to use the find my location function.

Figure 31. The feedback form for user to fill in any feedback and send to the developer team.
Appendix III – List of Database Tables

There are 17 tables in the database as listed below

Figure 32 The list of all tables in PostgreSQL

Figure 33 Table "feedback"

Figure 34 Table "mtr_exit"

Figure 35 Table "osm_nodes"

Figure 36 Table "osm_relations"
Figure 37 Table "osm_way_classes"

Figure 38 Table "osm_way_types"

Figure 39 Table "planet_osm_line"

Figure 40 Table "planet_osm_point"
Table “relations_ways”

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>relation_id</td>
<td>bigint</td>
<td>plain</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>way_id</td>
<td>bigint</td>
<td>plain</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>character varying(200)</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Foreign-key constraints:
- `relations_ways`relation_id_fkey FOREIGN KEY (relation_id) REFERENCES osm_relations(relation_id)

Has GIDs: no

Figure 41 Table "relations_ways"

Table “planet_osm_polygon”

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>osm_id</td>
<td>bigint</td>
<td></td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highway</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>amenity</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tourism</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>station</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>leisure</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shop</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>railway</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>public_transport</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>toilets:wheelchair</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair:description</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair:toilet</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>way</td>
<td>geometry(Geometry,3857)</td>
<td>main</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "planet_osm_polygon_index" gist (way) WITH (fillfactor='100')

Has GIDs: no

Figure 42 Table "planet_osm_polygon"

Table “planet_osm_roads”

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>osm_id</td>
<td>bigint</td>
<td></td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highway</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>amenity</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tourism</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>station</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>leisure</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shop</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>railway</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>public_transport</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>toilets:wheelchair</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair:description</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheelchair:toilet</td>
<td>text</td>
<td>extended</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>way</td>
<td>geometry(LineString,3857)</td>
<td>main</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "planet_osm_roads_index" gist (way) WITH (fillfactor='100')

Has GIDs: no

Figure 43 Table "planet_osm_roads"

Table “route_bus”

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_id</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>company_code</td>
<td>text</td>
<td></td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>district</td>
<td>bigint</td>
<td></td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>route_name</td>
<td>text</td>
<td>not null</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>route_type</td>
<td>bigint</td>
<td></td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service_mode</td>
<td>text</td>
<td></td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>special_type</td>
<td>bigint</td>
<td></td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loc_start_name</td>
<td>text</td>
<td>not null</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loc_end_name</td>
<td>text</td>
<td>not null</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "route_bus_pkey" PRIMARY KEY, btree (route_id)

Has GIDs: no

Figure 44 Table "route_bus"
### Table "public.route_stop"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>route_id</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>route_seq</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop_seq</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop_id</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop_name</td>
<td>text</td>
<td>not null</td>
<td>extended</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Has OIDs: no

### Table "public.spatial_ref_sys"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srid</td>
<td>integer</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auth_name</td>
<td>character varying</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>auth_srid</td>
<td>integer</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>srtxt</td>
<td>character varying</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>projtext</td>
<td>character varying</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)

Check constraints:
- "spatial_ref_sys_srid_check" CHECK (srid > 0 AND srid <= 998999)

Has OIDs: no

### Table "public.stop_bus"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lat</td>
<td>double precision</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lon</td>
<td>double precision</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop_id</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop_type</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>bigint</td>
<td>not null</td>
<td>plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>geom</td>
<td>geometry</td>
<td>not null</td>
<td>main</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Has OIDs: no

### Table "public.ways"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
<th>Storage</th>
<th>Stats target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gid</td>
<td>bigint</td>
<td>not null</td>
<td>default</td>
<td>nextval('ways_gid_seq'::regclass)</td>
<td>plain</td>
</tr>
<tr>
<td>class_id</td>
<td>integer</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length_m</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>text</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source</td>
<td>bigint</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target</td>
<td>bigint</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reverse_cost</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost_s</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reverse_cost_s</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rule</td>
<td>text</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>one_way</td>
<td>integer</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxspeed_forward</td>
<td>integer</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxspeed_backward</td>
<td>integer</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>osm_id</td>
<td>bigint</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source_osm</td>
<td>bigint</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target_osm</td>
<td>bigint</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority</td>
<td>double precision</td>
<td>not null</td>
<td></td>
<td>default 1</td>
<td></td>
</tr>
<tr>
<td>the_geom</td>
<td>geometry</td>
<td>not null</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indexes:
- "ways_pkey" PRIMARY KEY, btree (gid)
- "ways_gid_seq" gseq (the_geom)
- "ways_osm_id" btree (osm_id)
- "ways_source_osm_id" btree (source_osm)
- "ways_target_osm_id" btree (target_osm)

Has OIDs: no

---

**Figure 45** Table "route_stop"

**Figure 46** Table "spatial_ref_sys"

**Figure 47** Table "stop_bus"

**Figure 48** Table "ways"
Figure 49 Table "ways_vertices_pgr"