FYP 16008
WheelGO - A Navigation System for Wheelchair Users

Interim Report

Supervisor:
Dr. T.W. Chim

Group Members:
Chau Shing Yi (2013712810)
Nip Chi Fung (2013571644)

Date of Submission:
22nd January, 2016
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 be 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.

The paper 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. 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.

Meanwhile, a prototype with a map embedded was released, and deployment of routing algorithm is followed. Moreover, implementation of audio navigation function and feedback mechanism will start after deploying the routing algorithm. 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

We would like to express our gratitude to 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

ABSTRACT .......................................................................................................................... 2
ACKNOWLEDGEMENT ................................................................................................. 3
LIST OF FIGURES ........................................................................................................... 5
LIST OF TABLES .............................................................................................................. 5
ABBREVIATIONS ............................................................................................................. 5

SECTION I – INTRODUCTION ......................................................................................... 6
  1.1 INTRODUCTION ..................................................................................................... 6
  1.2 EXISTING SOLUTIONS ....................................................................................... 7
  1.3 OBJECTIVES ......................................................................................................... 9
  1.4 TECHNICAL OBJECTIVES ................................................................................ 9
  1.5 SCOPE .................................................................................................................. 10

SECTION II – APPROACH AND METHODOLOGY ...................................................... 11
  2.1 SOFTWARE DEVELOPMENT CYCLE ................................................................... 11
      2.1.1 Design Phase ............................................................................................... 11
      2.1.2 Implementation Phase ................................................................................. 11
      2.1.3 Testing Phase ............................................................................................. 11
  2.2 DEVELOPMENT COMPONENTS .......................................................................... 12
      2.2.1 Equipment Setup ....................................................................................... 12
      2.2.2 Development Tools ................................................................................... 12
      2.2.3 Infrastructure Overview ............................................................................. 13
  2.3 TASK LIST, DIVISION OF WORK AND STATUS ............................................... 14

SECTION III – PROGRESS REPORT ............................................................................ 15
  3.1 PROGRESS SUMMARY ....................................................................................... 15
      3.2 DESIGN ............................................................................................................ 15
      3.2.1 Database Design .......................................................................................... 15
      3.2.2 User Interface Design ................................................................................ 16
      3.2.3 Routing Algorithm Design ......................................................................... 16
      3.2.4 Vocal Guidance Design .............................................................................. 16
      3.2.5 Test Data and Test Case Design ................................................................ 16
  3.3 IMPLEMENTATION ............................................................................................... 17
      3.3.1 [Milestone 1] Building the MapView through external Tile Server provider .. 17
      3.3.2 [Milestone 2] Routing .................................................................................. 18
  3.4 FUTURE WORK PLAN ............................................................................................ 23

SECTION IV – CONCLUSION ......................................................................................... 24

SECTION V – REFERENCES ............................................................................................. 25

SECTION VI – APPENDIX I ............................................................................................ 26

SECTION VII – APPENDIX II .......................................................................................... 27
List of Figures
Figure 1 A screenshot of the route suggested by the Hong Kong Society for Rehabilitation website. 7
Figure 2 Two paths from Knowles Building, HKU to Meng Wah Complex, HKU suggested by Google Maps, both paths include staircases. 8
Figure 3 Prototype demonstrating the features of WheelGO 9
Figure 4.1 Logo of Android Studio 12
Figure 4.2 Logo of PostgreSQL 13
Figure 4.3 Logo of QGIS 13
Figure 4.4 Logo of Mapbox 13
Figure 5 Infrastructure of the software 13
Figure 6 List of tables designed during designing phase 15
Figure 7 Interaction between WheelGO and tile server Mapbox 17
Figure 8 Graph with each vertex represents a geographical location 18
Figure 9 The database tables with the corresponding attributes 18
Figure 10 Interaction between WheelGO and the application server and database 19
Figure 11 Sample route without the staircases in application 20
Figure 12 (Left) The original OSM data with staircases (Right) The database has staircases removed (highlighted in red circle) 20
Figure 13 Raw data of bus stop. The names of the stops are missing and the coordinate system used is different from that of OSM. 21
Figure 14 Visualization of a route with QGIS 21
Figure 15 Illustration of the interaction between mobile applications and the Mapbox web server. 22
Figure 16 Two paths returned by the Mapbox API 22
Figure 17.1 UI for displaying turn-by-turn routing function 27
Figure 17.2 UI for the voice guidance function 27
Figure 17.3 UI for displaying accessibility function on the map 27
Figure 17.4 UI for the voice guidance function 27

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
Table 3 Summary of the progress of the 4 stages in implementation phase 24
Table 4 Test areas and passing criteria for the testing phase 28

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

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 scopes of the application will be illustrated in detail. After that, the methodology of the project and a brief progress report with the difficulties encountered will be listed out. Based on the work done and the problems encountered at the current stage, the future work plan of the project 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 deviates 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 imports the street view from Google street view and provides valuable information such as the accessibility of each facilities and their contact numbers. Since 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 talk full advantage of it because in some situations, all the routes recommended includes waypoints that are not suitable for them. Figure 2 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.

![Google Maps screenshot](image)

*Figure 2 Two paths from Knowles Building, HKU to Meng Wah Complex, HKU suggested by Google Maps, both paths include staircases.*

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*
1.3 Objectives

The project \textit{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 3). 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 \textit{WheelGO}.

1.4 Technical Objectives

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

- **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.

- **100% Routing accuracy without staircases**
  Routes computed by the application should be able to avoid staircases in the dataset.
1.5 Scope

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 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.

<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

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.

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.

The application will have 2 layers: cross districts layer and detailed map layer

- Detailed map layer: Indicates the accessibility of the places (e.g. restaurants/shopping malls) and shows the point-to-point shortest route
- Cross districts layer: Shows the point-to-point shortest route across different districts and suggests transport means

1.5 List of features

- Recommends routes for wheelchair users (limited to Hong Kong area)
- Displays turn-by-turn routing instructions
- Indicates the accessibility information of each point of interest
- Provides voice guidance and notification function
- Allows users to report bugs via feedback form
Section II – Approach and Methodology

2.1. Software Development Cycle
The project will 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 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 has started at early November, where the features and functions designed is implementing into the WheelGO.

2.1.3 Testing Phase
The testing phase is going to be carried out at late March. Testing will undergo a series of tests that test all components of the system as well as test for the precision of the routing algorithm.
2.2 Development Components

2.2.1 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)

2.2.2 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 allow create, edit, visualise, analyse geospatial information.

- **Mapbox**
  Mapbox is mapping platform that allow users to visualizing map data and do analysis in real time. It provides API (Application Programming Interface) for developers to design their own map in Mapbox Studio.
2.2.3 Infrastructure Overview
The following figure demonstrates the infrastructure of the whole software system. The application is connected to the application server and external tiles server. The database is connected through application server.

Figure 5 Infrastructure of the software system
### 2.3 Task List, Division of Work and Status

<table>
<thead>
<tr>
<th>Time</th>
<th>Task</th>
<th>Division of Work</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 2016</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September</td>
<td>Researched on the existing products on the market</td>
<td>Sabrina Chau, Terry Nip</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Investigated the feasibility of implementation different functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defined the project scope</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Designed the project webpage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prepared the project plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acquire the necessary development tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Brainstormed any additional value-added features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October</td>
<td>Build app with basic routing function</td>
<td>Terry Nip</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and input data into the map database</td>
<td>Sabrina Chau</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Draft the tentative UI</td>
<td>Sabrina Chau</td>
<td></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>
<td></td>
</tr>
<tr>
<td>December</td>
<td>Implement the turn-by-turn routing function</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td>Year 2017</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>January</td>
<td>Continue implementing the turn-by-turn routing function [Milestone 2]</td>
<td>Sabrina Chau, Terry Nip</td>
<td>Working</td>
</tr>
<tr>
<td></td>
<td>Write the interim report</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td>February</td>
<td>Implement acoustic navigation function [Milestone 3]</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td>March</td>
<td>Implement the feedback mechanism [Milestone 4]</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improve the UI design of the app</td>
<td>Sabrina Chau</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improve the routing algorithm in term of user experience</td>
<td>Terry Nip</td>
<td>Not Start</td>
</tr>
<tr>
<td></td>
<td>Test and debug</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td>April</td>
<td>Finalize the implementation [Milestone 5]</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Write final report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May</td>
<td>Prepare the exhibition poster</td>
<td>Sabrina Chau, Terry Nip</td>
<td></td>
</tr>
</tbody>
</table>
Section III – Progress Report

3.1 Progress Summary
This section demonstrates the work progress of the project. The team have accomplished the design tasks on schedule. The implementation of the application WheelGO has begun in early November. In the implementation stage, the team has set up four milestones, namely, map display, routing, voice guidance and feedback form. These functionalities will be implemented to the application incrementally. In current stage, the team has finished the implementation of the map display and is developing the routing algorithm for the wheelchair users.

The following subsections will demonstrate the progress and the challenges encountered in each stage in detail.

3.2 Design
3.2.1 Database Design
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 four 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. Barriers – The places which are inaccessible for the wheelchair such as staircase and slope with high gradient
4. 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 documents in the same database. Figure 6 shows the five tables designed in the database.
3.2.2 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. Tentative user interface can be seen in the appendix.

3.2.3 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.

3.2.4 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.

3.2.5 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 II.
3.3 Implementation

3.3.1 [Milestone 1] Building the MapView through external Tile Server provider

3.3.1.1 Overview

In Android application, MapView refers to a view component which displays a map. The first step for developing navigation system is to implement the appropriate MapView. Based on this map, the team can further develop related map features and functions. Google Maps API and OSM-based development platform Mapbox API are some of the tools that provide easy implementation of MapView. After extensive studies on the APIs (Detailed description of the challenges encountered using different map libraries can refer to Section 3.3.2.2), Mapbox is adopted to render the map tiles on the MapView on-demand. On application start, WheelGO connects to Mapbox server using Mapbox API.

![Mapbox API](image)

**Figure 7 Interaction between WheelGO and tile server Mapbox**

3.3.1.2 Challenge and mitigation

**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, and the working principles of our routing function are 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 approaches 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, where adding this feature will definitely benefit the users. Therefore, other Map APIs were considered for developing our application.

For this project, as the team defines it own routing algorithm based on OSM map system, therefore the team was decided to use Mapbox, a OSM-based platform as the tile server to render the map. Apart from its OSM-based nature, Mapbox also provides powerful functions such as manage map style, set and move camera position, add markers, and GPS location finding.
3.3.2 [Milestone 2] Routing

3.3.2.1 Developing the routing algorithm

The implementation of the routing algorithm for wheelchair users is still under development. Many studies on routing algorithms and APIs were carried out to obtain the optimal routing solution for our application. The working principle of the routing algorithm for wheelchair users is worth discussion. The two possible approaches are (1) applying self-defined Dijkstra’s Algorithm, and (2) obtaining paths from the Direction API of the existing map providers.

The first solution requires extra storage of graph component. Figure 8 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 optimized by adjusting the graph component since the graph is defined by the team.

The second solution obtains routes by using the existing APIs such as Mapbox. After receiving a HTTP request with coordinates, the server will return the routes in JSON format. The returned JSON object contains detailed route information including duration, distance, and step. The routing algorithm for wheelchair will be constructed by indicating and eliminating infeasible ones from those returned paths. After consideration, the team decided to implement its own algorithm. The reasons behind are discussed at 3.3.2.4.

3.3.2.2 Data Processing and Database configuration

To build the wheelchair-enabled routing function based on an optimized dataset, the team has set up a map database. The map database is created using PostgreSQL. Figure 9 shows the eleven tables and the corresponding attributes currently defined in the database. In the database, the tables can be categorized into 3 types, namely, OSM raw data, public transportation data, and routing data. OSM raw data is responsible for displaying accessibility information; public transportation data is required in calculating the routes with public transportation; routing data is needed for calculating the shortest route in normal (walking) mode. These tables are useful for routing and displaying accessibility information in the following steps.
3.3.2.3 Implement self-defined routing function

To realize the flexible routing function, an application server and the database has been put on Amazon Elastic Compute Cloud (EC2). The application server is written in node.js, which responds to HTTP routing request and connect to the database server. The database use PostgreSQL to store the tables. With the servers, whenever the user posts a routing request or intended to get accessibility information of any point on the map, the application send HTTP requests to the amazon EC2 with specific token so that after received the request, amazon EC2 redirects to the node.js application for processing. After process the query on the database, the application returns the HTTP responses in JSON format.

![Diagram](image)

**Figure 10 Interaction between WheelGO and the application server and database**

**Routing Algorithm Steps**

The routing algorithm is depends on the nodes and edges exists on the routing table. The following gives an example on the real implementation of routing algorithm:

1. Find nearest edge using Nearest Neighbour Search by sending a query to the database table

   **Query**
   ```sql
   select x1 as lon, y1 as lat ,x2, y2, source as id, target, the_geom as geom1,
   ST_Distance(ST_GeomFromText('POINT(114.1269824 22.3605564)',4326), the_geom) as distance
   from ways order by distance limit 1
   ```

   **Sample Response**
<table>
<thead>
<tr>
<th>lon</th>
<th>lat</th>
<th>x2</th>
<th>y2</th>
<th>id</th>
<th>target</th>
<th>geom1</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.1265131</td>
<td>22.350228</td>
<td>14.1270791</td>
<td>22.3507653</td>
<td>69222</td>
<td>52021</td>
<td>0102000...</td>
<td>12077e-005</td>
</tr>
</tbody>
</table>

2. Find nodes in database tables for sources and destination

   **Query**
   ```sql
   SELECT x1 as lon, y1 as lat ,x2, y2, source as id, target, the_geom as geom1,
   ST_Distance(ST_GeomFromText('POINT(114.1269824 22.3605564)',4326),the_geom) as distance
   from ways order by distance limit 1
   ```

   **Sample Response**
   ```json
   {"type":"Point","coordinates":114.126923927517,22.3606067}
   ```
3. Routing using Dijkstra's Algorithm with the nearest neighbours of source and destination

- Find the geometry information of each intermediate turn of the route

**Query**

```
SELECT lat, lon, ST_AsGeoJSON(the_geom) as the_geom FROM pgr_dijkstra( 'SELECT gid, source, target, cost, reverse_cost FROM ways', " + source_id + ", " + dest_id + " ), ways_vertices_pgr WHERE node = id order by path_seq;"
```

- Find the aggregate geometry information of the route

**Query**

```
SELECT ST_LineMerge(ST_AsGeoJSON(ST_Collect(the_geom)) as geometry FROM pgr_dijkstra( 'SELECT gid as id, source, target, cost, reverse_cost FROM ways', " + source_id + ", " + dest_id + " ), ways_vertices_pgr WHERE edge = gid;"
```

**Sample GeoJson Response**

```
{  
   "routes": [  
      {  
         "geometry": {  
            "type": "LineString",  
            "coordinates": [  
               [ 114.1666103, 22.3328049 ],  
               [ 114.1665388, 22.3327264 ],  
               ...  
            ],  
            "legs": [  
               {  
                  "steps": [  
                     {  
                        "geometry": {  
                           "type": "Point",  
                           "coordinates": [ 114.1406722, 22.2840877 ]  
                        },  
                        "maneuver": {  
                           "location": [ 22.2840877, 114.1406722 ]  
                        }  
                     }  
                  ]  
               },  
               ...  
            ]  
         }  
      }  
   ]
```

4. Displaying the routing information

The prototype is now able to return routes without staircases (see Figure 11). This is done by not importing map data with tag “steps” (equivalent expression to “staircases” in OSM tags). Figure 12 illustrates the differences between raw OSM data and data in our database.
3.3.2.4 Challenge and mitigation

Data pre-processing problem

In collecting the raw map and transportation data from the Internet to create our own database, there are a few issues in dealing with the raw data. First, the raw data are scattered. The raw data is spread over the Internet. For example, the general map data is collected from the OSM website, the bus stop positions are collected from the government database data.gov.hk. Second, the dataset is too large. It is difficult to do operation on top of large amount of data. Third, the raw data is of different formats. e.g. the file format of MTR data is .mdb, while the Bus data is stored as .csv. Besides, the coordinate system used by bus stop is HK 1980 Grid System (represents the geographic information as x, y coordinates as shown in the figure 13) while the OSM data uses geographic coordinate system (represents the geographic information as longitude and latitude). It is time-consuming to do conversion of each raw data before integrating all the raw data into the database.

Route visualization problem

The application adopts the approach of first locating the nearest source and destination before running the Dijkstra’s algorithm, however, this may sometimes results in a long distance between the path displayed and the source and end points input by the users if the paths are plotted directly. The blue point in Figure 14 is the user input as source, while the orange point intersect red and blue line is nearest neighbor exists on the routing table.

Work has been done to minimize the gap (to prolong the path to reach the source/destination from users from the orange point to red point). For instance, points of intersection (POI) between the source/destination and their closest edge are calculated. The two resultant points (red point is one of the points) are added to the head and the tail of the path respectively. Finally, the algorithm connecting the POI with the calculated route and plot the polyline against the path coordinates.

Figure 13 Raw data of bus stop. The names of the stops are missing and the coordinate system used is different from that of OSM.

Figure 14 Visualization of a route with QGis
Mapbox Direction APIs are not preferable
Mapbox is powered by the open data obtained from OSM. Figure 15 illustrates how mobile application obtains route information from the Mapbox web server. After receiving a HTTP request with coordinates, the server will 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. Figure 16 demonstrates 2 example paths returned by the Mapbox API. However in reality, only 1 path was returned for most of the test cases. Problems encountered when applying second routing approach mentioned in section 3.3.2.2. with Mapbox direction API. 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.
3.4 Future Work Plan

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map Display</td>
<td>Finished</td>
<td>Late November</td>
</tr>
<tr>
<td>Routing for wheelchairs</td>
<td>In progress</td>
<td>Late January</td>
</tr>
<tr>
<td>Voice guidance</td>
<td>Not Yet Started</td>
<td>Mid February</td>
</tr>
<tr>
<td>Feedback form</td>
<td>Not Yet Started</td>
<td>Mid March</td>
</tr>
</tbody>
</table>

Table 3 Summary of the progress of the 4 stages in implementation phase

To summarize the project status, the team has finished the implementation of the map display and is currently deploying the routing algorithm for the wheelchair users. Table 2 shows the summary of the progress of 4 stages in the implementation phase. The expected deadline for deploying the routing function for wheelchair users is in late January, and the implementation of voice guidance and feedback form will begin in February.

If the implementation stage is completed ahead of schedule, the team will consider further working on a more challenging problem, namely, the optimal path algorithm where “optimal” means the balance between finding “short path” and a “comfortable” path. The algorithm should fuse the geo-positioning data, historical data find out the favourite paths wheelchair users uses, and other accessibility information (such as the road texture, slope gradient) to determine the best route. The data will be combined to calculate the optimal route considering their accessibility, capabilities and profiles.
Section IV – Conclusion

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 be able to follow such as the paths that contain staircase. To effectively 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 accuracy of the GPS signal and outdated 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.

In order to improve the quality of the routes suggested, two possible steps could be done in the future work. First, the platform should be built as soon as possible to accommodate users’ feedback and do modifications. Second, more studies on the design of routing algorithm are need.
Section V – References


Section VI – Appendix I
The figures are the tentative user interfaces designed for WheelGO.

Figure 18.1 UI for displaying turn-by-turn routing function

Figure 17.2 UI for the voice guidance function

Figure 17.3 UI for displaying accessibility function on the map

Figure 17.4 UI for the feedback form of the feedback mechanism
Section VII – Appendix II

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*