Software Engineering Project COMP8790

College of Engineering and Computer Science

The Australian National University

Android Taxishare Client Application Development Report

YingYi Chen (u4704324)

Date submitted: June 4, 2010

Abstract

CSIRO employees are over all Australia. The employees often have business trips by domestic flight in Australia. Multiple taxis are used by CSIRO employees from an airport to the same CSIRO sites or events. Sharing same cab or rental vehicle with other employees travelling to the same final destination becomes an issue. To solve this issue, CSIRO develops a system called Taxishare system as a CSIRO internal service.

In order to improve user experience on Taxishare system, Taxishare client application is designed to provide a rich and friendly graphical user interface and a touch-screen control for end users on Google Android mobile platform. Within a tight development schedule, software engineering and project management techniques involve in application development life-cycle to solve issues and mitigate risks. Multiple design choices apply in different development iterations. A solution is selected by comparing different user interface designs and different architecture choices to achieve final Taxishare client application project objective. Limitations of Taxishare system and Taxishare client application are discovered by development team during the Taxishare client application development. Suggestions of improvement and enhancement on further development of Taxishare system are presented in the report.

1. Introduction

CSIRO provides taxi sharing service called Taxishare System for all CSIRO employees to share a cab or rental vehicle with other colleagues during business trips. There are multiple interacted interfaces with taxi sharing service including SMS interface, Twitter interface and mobile application interface. Android Taxishare client application is client side software for taxi sharing service. The android Taxishare client application allows the employees to interact with taxi sharing system easily and effectively on Google Android Platform.

The primary objective of Taxishare client application project is to deliver a client side software for taxi sharing system which provides map a view interface and a multi-touch control on Google Android Platform within 3 months schedule. Another aim is to demonstrate and apply appropriate software engineering and project management techniques to achieve the project goal in Taxishare client application development process.

In the report, multiple interface designs and different architecture decisions in different development phase are presented. The report describes each of single development iteration under iterative and incremental development life-cycle. Each of development iteration discusses about resolving issues, user experience on interface and control, comparison of different design choices, usage of Google map, Google geocoding, GSP and relevant techniques in Android Taxishare client application. In addition, the report proposes improvements and enhancements on further Taxishare system and Taxishare client application development.

2. Background of Taxishare System

2.1 Taxishare System Project Development

Originally Taxishare system intends to assist CSIRO employees to solve a sharing taxi problem inside CISRO community. CSIRO employee often take a cab or rental vehicle individually without sharing with other CSIRO employees who travelling to same final destination, It is necessary to share a taxi to same destination rather than taking multiple taxis individually. Specially, for CSIRO employees who need to have

business trips by domestic flight in Australia frequently, they often need to take a taxi from an airport to CSIRO events when they arrived at an airport. The idea of sharing taxi can reduce cost of travelling and save time from waiting queues of airport for CSIRO employees. From environment protection aspect, the carbon footprint is reduced along with the number of taxi decreased. Same CSIRO employees have an opportunity to collaborate and be familiar with each other via sharing taxi. It is convenience for CSIRO employees to share taxi. They only need to provide a final destination to the taxi sharing system. The taxi sharing system interpret the addresses and help the employees contact with other employees who travelling to same final destination. The original taxi sharing system is planned to use daily as internal service for CSIRO employees. In early concept of taxi sharing system, the starting point of taxi sharing system is limited to an airport only. The system allows employees to interact via phone, email, SMS or web form (pre-registrations). The system receives the request and matching with colleagues who have a similar arrival time and final destination. The system responds matching result message with colleagues' contact number to the requester. The employees can call the other employees and arrange to meet them in the taxi queue or at the Taxi sharing meeting point. Any pre-registrations users can monitor the service on your arrival through web service.

Taxishare system project is supervised by Dr. Ken Taylor. The System is implemented by Herman Jaya who is an industrial trainee of CSIRO. The project is launched on Wednesday April 2009. Advance preparation of Taxishare system project includes engaging a student trainee, establishing project team, confirmation of project budget which takes two months. The actual project development starts from Jun 2009. The project plans to finish on March 2010. Due to development schedule of Taxishare system is tight and the project is lack of developer in the project development. The project requirement and design documentation is not explicit. Taxishare system is implemented iteratively. Different enhancements and new requests are added in each of iterative development after the Taxishare system framework is created.

Basic design framework of Taxishare system is that two people send SMS requests with exact same final destination to Taxishare system. The system matches the incoming requests and notifies the requesters by sending respective phone numbers to both of the requesters. If the requests still no match within 15 minutes, the request

will be discarded by the system. The use case assumes that default of starting point is airport. The first phase development does not include advance matching algorithm so that the miss spell address names or ambiguous address names are not able to match. The simple framework of Taxishare system is created in this phase.

More new features and improvements are added in second phase development. In order to avoid miss spelling address typed by users, SOUNDEX matching algorithm is applied in Taxishare system matching function. The cancel function is implemented so that users can cancel the service if they do not want to wait for a request from the system. The restriction of starting point is removed. The users are able to submit any location as a starting point to Taxishare system. The system also adds constraint of number of message to prevent messages flood from a same person. Two matching mechanisms are proposed. Both mechanisms are able to handle more than two requesters. One mechanism is that the system groups all requesters travelling to same destination in one rental vehicle. Another one is that all requesters are assigned into pairs. The idea of grouping all the requesters is denied eventually.

After the back end service logic of Taxishare system with simple SMS interface is constructed, web interface of Taxishare system start to be developed. The web interface with an online map is a visual interface to show all the request and matching result. The interface intends to be placed as an airport advertisement so that all users are able to view all sharing taxi requests and arrange their own meeting points in airport .There is no ability for web interface to contact with Taxishare system service. The web interface only displays SMS data and shows the request location people heading to in Google map. All SMS requests received by Taxishare system will be pinned on the Google map. The web based interface is categorised into different cities. There are ten cities Darwin, Brisbane, Canberra, Sydney, Melbourne, Caims, Hobart, Perth, and Newcastle in Australia available for web based interface.

The direction of Taxishare system development switched to interface development of Taxishare system after Taxishare system service is available. Currently, SMS interface, web interface and Twitter interface have been developed by developer team. More interfaces are planning to develop for Taxishare system including Tripit interface, Email interface. All those interface increase the usability of Taxishare system service. It is convenience for users to access taxi sharing service via multiple interaction

method. However, the back end system logic is enhanced and improved slowly. With growth of number of Taxishare system interface, more user interfaces with advance features are implemented in Taxishare system. The Taxishare system design and logic fall behind. More defects of back end system are discovered during the user interface test. The original matching algorithm for SMS interface is not longer suitable for advance interface. The back end server supported for advance user interface needs to be improved in further development.

2.2 Taxishare System Architecture and Process Flow

The Taxishare system with multiple user interfaces provided full functional taxi sharing system service for users. The **Figure 1** illustrates architecture of the Taxishare system with all interface communication channels. The process flow is triggered by a request SMS from users. Users send SMS to SMS service provider. The SMS message need to base on either ' FROM TO AND ' or 'ONLY TO' format. In 'FROM TO AND' format, 'FROM' value must be set as start location and 'AND' must be set as destination location. In 'ONLY TO' format, default start location is set as airport, only destination location needs to be provided in SMS request. Taxishare system receives SMS request via HTTP from an SMS provider. The system extracts the location name from SMS request then than validation process is applied by the system. The process flow will be stopped by the system, if same number with same address request exist in database or sender has already sent more than 10 SMS in a quarter hour, more than 10 SMS per hour or more than 200 SMS per day. Meanwhile, system attempt to interpret the exact place name via Google MAP and stores the top listed place name into database. Once process flow completes successfully, the incoming SMS request will be stored in database waiting for matching process. SOUNDDEX matching algorithm is involved to mitigate incomplete address error. If SOUNDDEX cannot solve the matching logic, the system will search location name from address lists database which construct from Google Map API. Finally, SMS notification with matching result will be sent to requester by the Taxishare system.

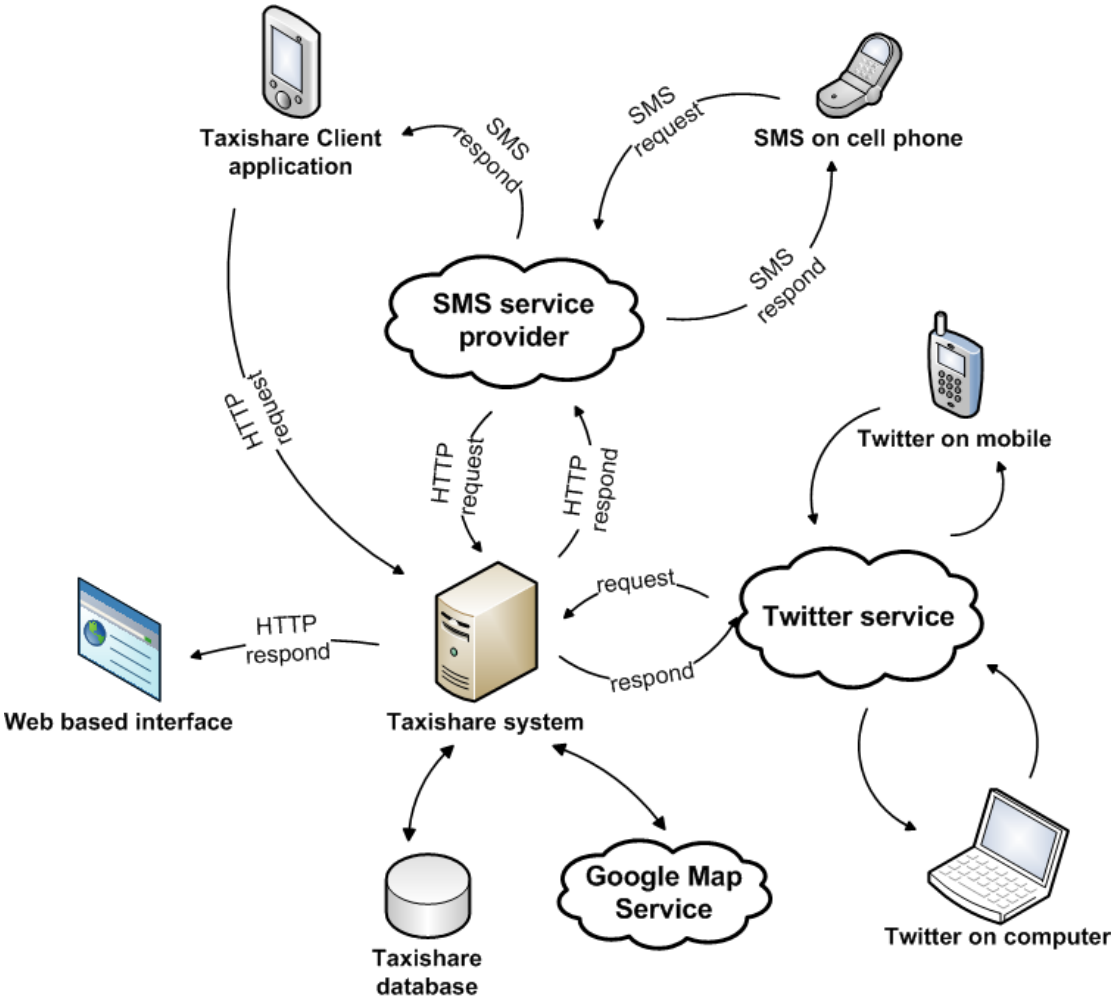

Figure 1: Taxishare system architecture interact with different communication interfaces

3. Android Taxishare Client Application Project

3.1 Introduction Taxishare Client Application

Taxishare system service offers more than one communication channels for CSIRO employees. There are SMS interface, web-based interface and Twitter interface interacting with Taxishare system service. All those interfaces exists different limitations in Taxishare system. Android Taxishare client application is a new communication channel for the system which intends to improve user experience and make Taxishare system effective and easy to use.

Comparing with other interfaces, SMS interface is a general option for CSIRO employees to contact taxi sharing service from the Taxishare system, due to all mobile

phone devices provide simple message service feature. Another advantage for SMS interface is that Taxishare system can easily receive precise contact number from incoming SMS messages. However, simple text interface might cause uncompleted input address or ambiguous location name by typing an address manually so that Taxishare system is hard to match with other requests precisely. For instance, two places in different cities have exact same location name. It is easy for Taxishare system to match those two places basing on same location name if those two addresses are not distinguished by names of the cities. Special text message format must be defined for the Taxishare system to extract location information from the text message which includes start pointing and destination addresses. The request message without following text format will be neglected by Taxishare system. SMS transmission service of Taxishare system is dependent on a third party SMS provider. The service quality of third party SMS provider determines message send and respond time. SMS delay issue occurred twice during SMS interface test. SMS responds from Taxishare system are delayed 45 minutes unexpectedly by the third party SMS provider. After the issue occurred on second time, Taxishare system development team decides to change to other SMS service provider.

Web-based interface designs to display all income and outgoing messages from Taxishare system. The user interface combines with Google map. All the request locations are marked by bubble pin on Google map and request messages including contact number and destination address are listed on left panel of interface at same time. CSIRO employees could view intuitively their requests from web-based interface after submitting taxi sharing request to Taxishare system. However, defect of web-based interface is that users are not able to control and interact with Taxishare system via web-based interface. Web-based interface only is regarded as an auxiliary tool for Taxishare system to verify incoming messages. Additionally, browsers on low performance mobile cannot support Google Maps. It is not convenience for users to submit request via mobiles to the system and then view the request from computer browser.

Twitter social networking is widely used in the world and Twitter interface provides multiple version interfaces for different mobile platform. Instead of developing new interface for Taxishare system, connecting Twitter service with Taxishare system as

one of communicated channel of Taxishare system which increases the usability of taxi sharing system. Entire taxi sharing process can be communicated by users via Twitter social networking. An advantage of social community over SMS communication is that users can view other users' profile and decide whether they wish to share taxi with them. CSIRO employees who has Twitter account not only can send sharing taxi request to Taxishare system via Twitter but also the request can be viewed from Twitter community by other CSIRO employees. However, Twitter interface is still text based interface so it has same limitations as SMS interface. In current Taxishare system, Twitter interface is not able to interact with SMS interface. Twitter interface only allows users to communicate within Twitter social networking. Because twitter request messages does not contain any contact number, Taxishare system are not able to extract contact number from request messages and send contact number back to any SMS requester.

A simple client application with online map view on mobile platform is designed for Taixshare system to eliminate or mitigate these defects from SMS interface, web-based interface and Twitter interface. Client application is decided to be implemented by development team on Google Android operating system for mobile platform. Android operating system distribution was announced on 5 November 2007, until now Android platform obtain 9.0% of smart phone market share and take 4th place in current mark share. With rapid growth of market share of Android platform, it is worth and necessary to develop a free client side application for taxi share system on Android platform. From development aspect, Google Android development community offers comprehensive support for developers including developer mailing list, user mailing list, API demo, development guide and developers' blog. All these resources help new developer pass the learning curve of android development and adapt into Android development environment quickly within three months project development schedule. The primary features of client application for Taxishare system is to build online map display and control into application and to provide feature to mitigate miss spelling or ambiguous input address. The Goolge Android system has an external Google Map library which allows developer to create dynamic map view and map control on Taxishare client application. Also Android development API includes Google geocoding library for Taxishare client application to interpret locations. All these complete supports make implementation of online map on

Taxishare client application within short term development become feasible.

Taxishare client application on Android platform uses dynamic Google map view to display locations for users. So users are able to control and move the map by drugging operation and use the zoom control to view the detail address easily. To avoid miss typing input issue which appears on text-based interface, The Taxishare client application have ability to mark the locations via different bubble pins on the map, instead of manually typing location name in text field. The application also allows users to type a location name in text interface manually. However, a list of suggestions of location name is proposed by the application, if there is any ambiguous address or incomplete address in text input. As a result of Taxishare client application establishes a connection with Taxishare system service via HTTP transmission directly. The delay factor of third-party service provider delay is excluded from the application. Also respond time from Taxishare system service is faster than Twitter interface and SMS interface. Comparing with SMS interface requires to SMS charge for using Taxishare system service, Taxishare client application offers completely free service for users via Internet connection. Android Taxishare client application is flexible to add new features for Taxishare system in future. For further development on Taxishare system, more advance features and enhancements will be implemented in Taxishare system. Android platform development provided rich API to adapt further change on Taxishare system.

3.2 Project Planning and Preparation

The three months development schedule of Android taxishare client application is intensive. The project is lack of human resource. There is only one developer in self-organise development team. The project requirement is implicit so it is possible to change the requirement at any time during project development. However, the development team is able to communicate with customers via face-to-face conversation to verify and confirm the requirements. Working software or component must be reported or demonstrated by project team periodically. Due to those factors, iterative and incremental development methodologies are applied for Android taxishare client application by development team to adapt quickly to changes of the project and present the portion of product progressively. The development planning is

obtaining the smallest workable piece of functionality to deliver simple project objective early and then the project follow iteration and incremental approach until full functional software is produced. In each of iteration, the milestone of development must be identified to avoid continue to change the requirements, because of customer demand. The project continually improves or adds new feature in the application throughout entire life of the project. This strategic plan ensures that Android Taxishare client application can be delivered on time with partial or full functionality. The requirement of android Taxishare application is break down into different portions. The sub portion of the application is carried out independently and iteratively. Final iteration integrates all the portion of the application into full functional application and test the full functional application.

Identifying and managing the risks on the intensive project before the project starts will reduce the failure of the project. From the project scope aspect, as a result of implicit requirements of Android taxishare client application, scope creep is measure as high probability risk. It is probable for customer to change project requirement as their demand during project development without defining clear project requirement. For iterative development lifecycle, there is not clear milestone in each of iteration to identify iteration goal. Developer team plans to contact with customer constantly and confirm the project requirement and design before each of iteration development. Both project resource and project schedule are fixed, there are only three months for project development including advance project preparation, project design, project implementation and final application test. Also Only one member is responsible for all project development jobs as self-organise team. All the risk related to resource and schedule is hard to avoid, however, shrinking the project objective can help mitigate those risk. Development team need to negotiate with customer to limit requirement implementation if the project development is out of schedule. At the beginning of project, wild ambition for the project often cause estimate error. It is necessary to analysis feasibility of each of design choices and identify the priority of the features proposed by customer. Iterative and incremental development methodologies ensure deliver workable software or component in each of development phase to avoid wild ambition project objective.

Technical issue of Taxishare client application on Android platform also need to

estimate during advance preparation of the project. The first technical issue is on obtaining the users' correct mobile number and pass to taxishare system. Mobile number does not store in local mobile phone device or mobile SIM card unless users manually input their mobile number into mobile system setting. As an example of using SMS service (see **Figure 2**), regularly, the message which is text only without any personal profile information from the sending mobile is delivered to a central short message centre (SMSC). After storing the message in SMSC, the mobile number and other information from the sending mobile will be search from a Home Location Register (HLR) which is a main database to store mobile subscriber information. The SMS text message with extra information including header, service centre timestamp, the phone number, protocal identifier, the length of the message will be sent by SMS centre to the recipient. Mobile device numbers are stored in database of Telecommunication Company. The design approach to obtain the users' mobile phone number is directly influence on communication architecture between the application and taxishare system. Android client application intends to interact with Taxishare system via HTTP transmission. It is free service through Internet and without providing any personal profile information. The security issues must be considered. Malicious flooding messages cause taxishare system crash easily. Until now, Android system announces seven version platforms in Android market. Platform compatibility consideration affects development libraries in Android application development. It is important to test Android taxishare client application on multiple versions of the platform and choose common implementation approach in order to successfully operate on different version of platform.

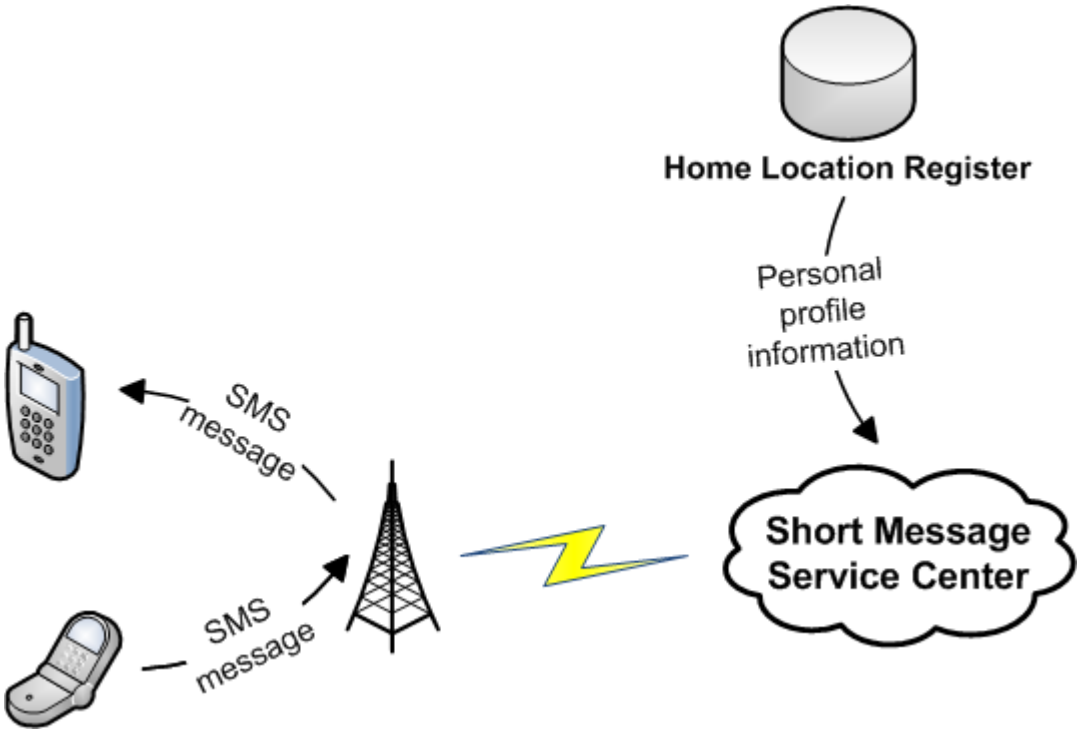

Figure 2: Demonstrate about SMS transmission to receive personal profile information from Home Location Register

Android development framework is generally forward compatible with new version of the Android platform. The application requires to Google map service. The application is developed on old version 1.5 Android platform. Application development libraries contain android 1.5 platforms API and Google 1.5 external Google Maps library. Google map service requires registering for debug certificate and signs a debug key for application for Android application development purpose. To publish Android taxishare client application, another formal registration is required by Google. In addition, all applications on Android platform are written using Java programming language. The Java Virtual machine must be installed on development environment. The application will debug on 1.5 Android platform mobile emulator and test on Google Nexus One mobile device running on 2.1 Android platform.

In summary, advance preparation of Taxishare client took one month before the project development. The technical preparation includes Android development framework studying, Taxishare system design and architecture studying and Android development environment configuration. For the project preparation, risk is identified and risk management plan is prepared, requirements and objectives of Taxishare client

application project are collected from the customer.

3.3 Android Taxishare Client Application Project Development

Android Taxishare client application project development uses iterative and incremental software development. The entire development lifecycle is divided into four iterations. Each of iteration produces workable components for Android Taxishare client application. The iteration contains common development process including requirement analysis, design, implementation and test. The first iteration produce basic Taxishare client application framework includes text based user interface, SMS interaction component, Http interaction component. In second iteration, Google map display and control development is involved. Development focuses on implementing different operations on map view. All components including interaction features, validation features are integrated with Google map in the third iteration. The primary task in final phase development is to test entire Taxishare client application with Taxishare system.

The First Iteration In the first development phase, the development started on solving retrieve mobile number issue. In order to obtain the mobile device number from Home Location Register database of Telecommunication Company, an approach was that Taxishare client application sends SMS validation message to Taxishare system via SMS service provider before establishing connection with Taxishare system via HTTP and then the application receive SMS acknowledgement message with sender's mobile number from Taxishare system. Taxishare client application retrieved mobile number from SMS acknowledgement message. A benefit of the process was that users do not need to manually input a mobile number into the application. However, Adding SMS validation approach made application operation redundant. The feature duplicated with SMS interface. The approach did not eliminate issues of SMS interface. So the application would have same SMS service provider dependency and respond delay issue as SMS interface. SMS service provider could delay the SMS validation message so that Taxishare client application could not start taxi sharing request without receiving mobile number of SMS acknowledgement message from taxishare system. Additionally, Taxishare system server needed to implement a component to process the validation message only for the client

application. The development effort and complexity of application were increased by adding this feature. The project goal is to provide efficient client application for Taxishare system and resolve the issue existing in Twitter, SMS and web interface. So the design was denied due to the approach conflicted with project objective and project schedule. The application allowed user to type or change mobile number in the application before starting a Taxishare request. The mobile number would be record into mobile storage so that user did not need to type mobile number every time.

The communication component of Taxishare client application was created in the first development phase. Android development library included Apache HTTP client library. It was easy to implement HTTP transmission in Android application. There were two design choices for client application to communicate with Taxishare system. One was HTTP request with SMS respond approach (see **Figure 3**), anther one was HTTP request with HTTP respond approach (see **Figure 4**). Matching result responds from Taxishare system could be delivered via either SMS or HTTP. For HTTP responds approach, the respond transmission design will be consistent with request transmission. SMS service provider limitation would not exist in the approach. Also the respond and request time were reduced by connecting with Taxishare system directly. Comparing with HTTP transmission, SMS respond approach was simple to implement, due to Taxishare system server supported for SMS interface. Reusing the existing SMS respond component in Taxishare system server reduced development effort. SMS responds could be handled by SMS application in mobile device which separate from Taxishare client application. So SMS responds approach decoupled client application and Taxishare system. HTTP request with SMS respond approach was stateless communication therefore the application did not need to keep running to waiting for a matching result. To implement HTTP respond approach in client application, a component required to be created in server side. The component is responsible for sending HTTP responds with matching result back to the application. In client side application, there must be a component building in the application to wait for HTTP responds from Taxishare system. The client application needed to keep running to trace Taxishare system matching result which was not convenience for users. This design had strong coupling between the application and Taxishare system. For the further development, Taxishare system HTTP request with HTTP respond approach was hard to maintain and modify. If any changes happened in either server

HTTP component or client HTTP component, both of them need to be changed consistently. Taxishare system required an external component to link SMS interface respond component with HTTP respond component. Otherwise SMS interface could not interact with client application on mobile. HTTP request with HTTP respond design approach caused extra workload for the project however design cannot produce better result. HTTP request with SMS respond approach was selected as final communication design with Taxishare system. The project spent two weeks time to achieve application communication implementation. In two weeks schedule, the development team not only analysis multiple design choices and implement the design but also study on Apache HTTP client implementation and SMS service on Android application.

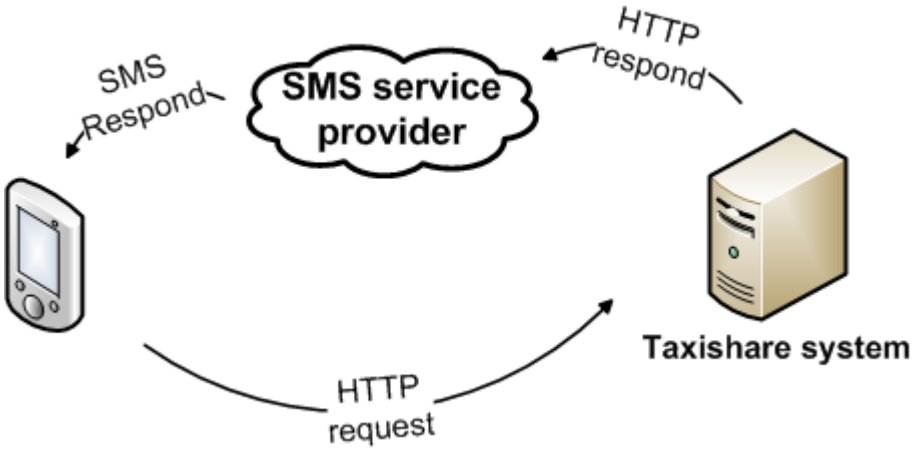

Figure 3: The application on mobile device sends a request via HTTP and receives a respond via SMS service

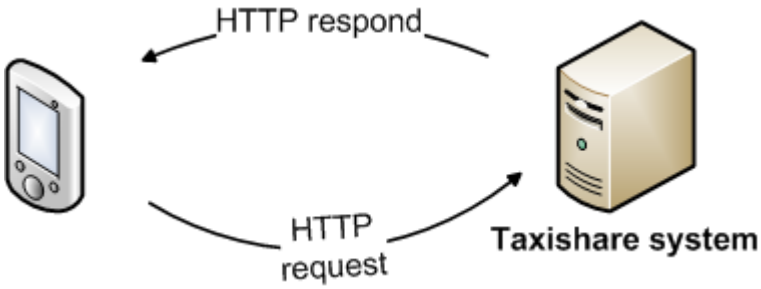

Figure 4: The application on mobile device sends a request via HTTP and receives a respond via HTTP.

The Second Iteration In the second iteration, text based interface was produced for

the application to test HTTP transmission. Also Google map interface and control was constructed as an independent component. Text based interface for the application was simply follow SMS interface input format. Text interface contained source text field and destination text field. However text based interface was hard to integrate with Google map interface and text based interface on the client application did not have significant changes over SMS interface. Eventually text based interface was stopped to develop in further iteration. The second phase of development focused on implementing Google map interface framework. The objective of developing application interface with Google map was to improve all aspects of user's interaction with existing Taxishare interface including perceived, learned and used. The client application interface design referred to official Google map application on Android mobile phone. Due to official Google map application is wildly used in mobile phone device. The graphical user interface of official Google map mobile application is well known for mobile users. Taxishare client application with same graphical user interface was easy for mobile users to identify Taxishare client application. All the application operation was simulated with official Google map application on Android mobile phone (see **Figure 5** and **Figure 6**). From the visual point of view, the Android Taxishare application had different mode of views which were satellite mode and normal mode. Normal mode provided simple and clear map view for user. Satellite mode had high-resolution satellite images for areas. So users could perceive different map views as their preferences. There were multiple navigation operations to control map in the Taxishare application. Regularly, users could drag the map view manually to find the location. Taxishare system application provided extra search operation to find a location from Google map via Google search bar. The application zoomed into the location on Google map automatically after typing the location name into Google search bar. The location would be focused on the centre of screen by the application. Geocoding service was applied in the search function to help reverse the location name to longitude and latitude coordinate. The application used these coordinates to locate the address on the application map. Instead of typing an address name manually in text base interface, marker function was designed to mark a location by dragging marker to on the top of the location and recorded the location name automatically into the application. Two marking operations were implemented by development team in mark function. The original mark operation was via double clicking on the map to mark the location. Due to android development framework did not support for double

clicking action and action event, development team spent half weeks on implementing double clicking action for the application. However double clicking operation was not adopted by the development team, double clicking operation was not wildly use in Android platform. Accessibility of the application was decreased by adding double clicking operation, because users might not be familiar with double clicking operation, it was hard for first-time user to understand intuitively control. The double clicking operation was replaced by dragging and dropping the mark on the top of the address action to mark the location. After the mark was located on the map, the application would call Google geocoding service to translate coordinate of the map into actual location name and display the address for users. From the project management aspect, the second iteration took 3 weeks development time. Multiple graphical interfaces and different interface control were designed and implemented during the second iteration. All the designs were demonstrated to customer and the application was adjusted by development team frequently following customer's feedback to achieve customer satisfaction. The major portion of graphical user interface was created in the phase. The project was under schedule until the second iteration finished. All the project risk did not become issue for project in the phase.

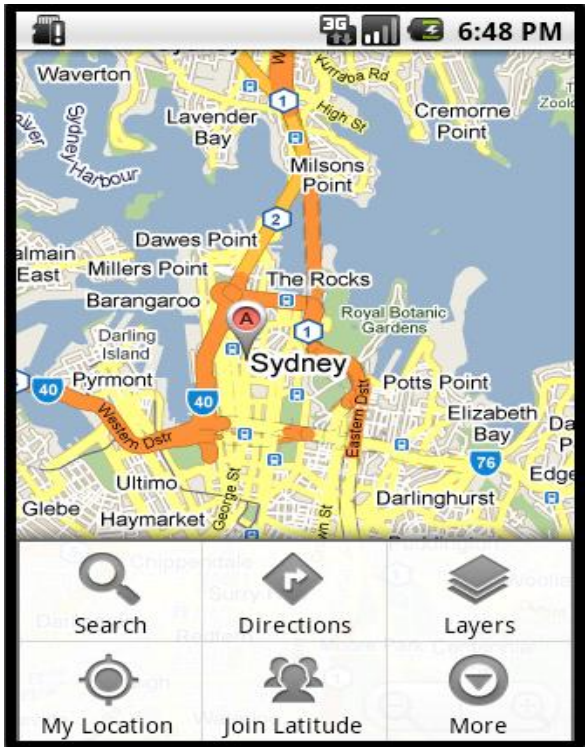

Figure 5: The screenshot of Official Google Map user graphical interface on Google Android platform

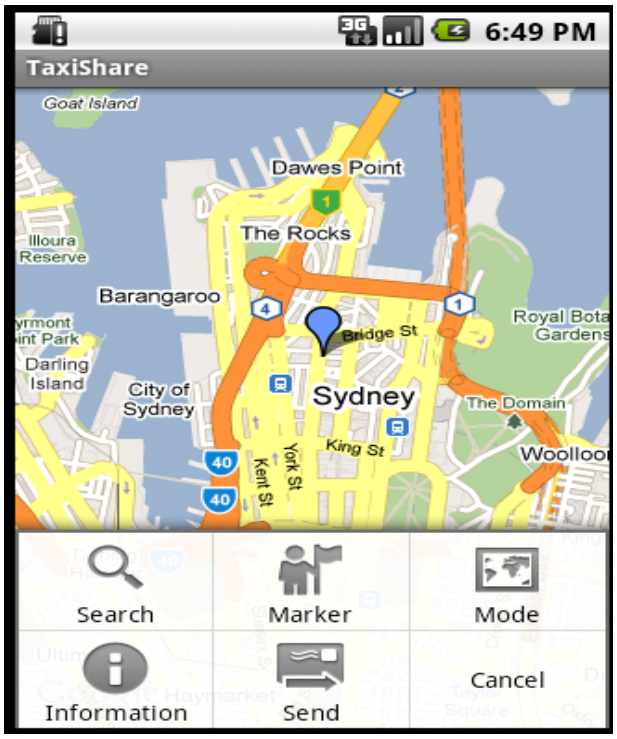

Figure 6: The screenshot of Taxishare client application user graphical interface on Google Android platform.

The Third Iteration Advance features and enhancements were added in user client application in the third phase. Developer team composed user graphical interface with all the components which were developed in previous phase in the iteration. All individual workable components were built together and arrange in one control screen so that users did not need to switch to different screen to operate different function. In addition, existing components were improved in the iteration. The location search bar was delivered in the second iteration development. To assist the location search bar to adjust ambiguous or miss spelling location name, the address suggestion mechanism was implemented for the location search bar. The users input with ambiguous or miss spelling location name was passed to Google geocoding service by the application. The geocoding service adjusted the miss spelling location name and returned the location suggestions list for users so the users did not need to concern about location typed error. HTTP interaction component was build in send and cancel function to communicate with Taxishare system. All the functions were grouped in a pop up option menu. The option menu provided all the features in Google Map so users could navigate all the operation easily. Internet connection checking mechanisms was developed to check the Internet connection on device also the mobile number input

component was added into the application before the application start. The iteration took two weeks in total. The android Taxishare application is completed at this stage.

The Final Iteration The aim of final iteration was to test Taxishare application communication. There was an issue on testing application in real environment. The Taxishare system service was in used by CISRO employees. Any test messages would mislead matching result in the system. Consideration of the flooding test messages might be influence on Taxishare service matching result and also there was budget cost on SMS provider service. Before deploy the application into Google Android Nexus mobile and test in the real device, the virtual test environment was created in one computer to simulate real world Taxishare system communication. All the Taxishare system source code and database were redeployed in local computer. A simple text-based web page was created to simulate as SMS interface. The test was simulated the communication between SMS interface represented by a simple web page and android Taxishare client application running in Android emulator. After fully testing on virtual environment, the client application tested on Google Nexus mobile. All the individual components were tested in each of iteration before integrating them into the application. So there was not any error or exception on Taxishare client application interface. Also the matching result between android applications was accurate. However, the matching result with SMS interface was out of expectation. The final iteration took 1 week in all. The application test risk did not estimate and identify in the project preparation phase. However, the Taxishare system had detail configuration and deployment instructions so that the testing issue was solved by creating a virtual test environment easily. Although the matching result with SMS interface did not completely satisfy original expectation, however, the Android client application interface still achieved customers' the requirement and satisfaction. The entire development life-cycle was terminated after complete the testing iteration.

3.4 Taxishare System and Android Client Application Issue

Due to Sounddex algorithm in Taxishare system server is a phonetic algorithm for indexing names by sound, as pronounced in English, Soundex algorithm only have ability to match despite minor differences in spelling. The algorithm is great for text based interface to adjust miss spelling input error. The client application has an ability

to provide full precise address including street name, street number, thoroughfare name, administrative area. However the Soundex algorithm cannot match simple address name with precise and complex address name. The **Table 1** shows testing data and matching result. Common string matching algorithm has a problem on matching ambiguous location name with precise location name for example, ANU string does not match with the Australian national university string.

Table 1: Inputs data of SMS interface and Inputs data of Taxishare client application interface, Also the matching result from Taxishare system.

SMS interface	Taxishare client application interface	Matching result
Childers city To Cowper Dickson	Childers city To Cowper Diskson	Match
Chliders city To Cwoper Dickson	Childer city To Cowper Diskon	Not Match
Childers city To Cowpre Dickson	Childers city To Cowper Diskson	Match
25 Childers street To Cowper Dickson	25 Childers street To Cowper Dickson	Match
Childers street To Cowper Dickson	Childers st To Cowper Dickson	Not Match
Canberra airport To Childers street	Canberra International airport To Childers street	Not Match
Airport To hombush	Airport to homebush	Match
Airport To homeboosh	Airport to homebush	Match
Airport To homebushes	Airport to homebush	Match
Airports To homebush	Airport to homebush	Not Match
Airport To Victoria	Airport to 120 Victoria street Melbourne	Not Match
Airport To Victoria	Airport to Victoria street Melbourne	Not Match
Airport To Victosria	Airport to Victoria	Match

Android Taxishare client application does not set up airport as starting point default value. The default start location value set up is missed in requirement preparation phase so that develop team does not implement this feature. There is not any mobile number verification feature in the application when mobile number is in wrong format.

The drawback of using Google map and Google geocoding in the application is that an Internet connection is required to get maps and related information from Google maps. If Internet transmission speed is slow, the map view display and location marking respond will delay.

3.5 Further Improvement and Enhancement

The advance matching algorithm shall replace existing matching algorithm in order to process with all kind of address name. The string matching algorithm e.g. Soundex algorithm is not appropriate algorithm for location matching. It could be use in adjusting the ambiguous or miss spelling location name only. For location matching, Taxishare system could calculate the longitude and latitude coordinate of locations to determine whether the distant of the locations is closed enough to match with each other. The Figure shows data flow of matching process. Taxishare system receives location names from SMS or client application communication channels. String matching algorithm e.g Sounddex algorithm adjusts those location names to reproduce location name in identical format. The Google geocoding web service interprets location name into the longitude and latitude coordinate. Taxishare system retrieves the location name with longitude and latitude coordinate. The distance among those locations can be calculated via longitude and latitude coordinates so that two locations with shortest distance are match. This matching design process has an ability to solve matching precise location name with simple location name issue (see **Figure 7**).

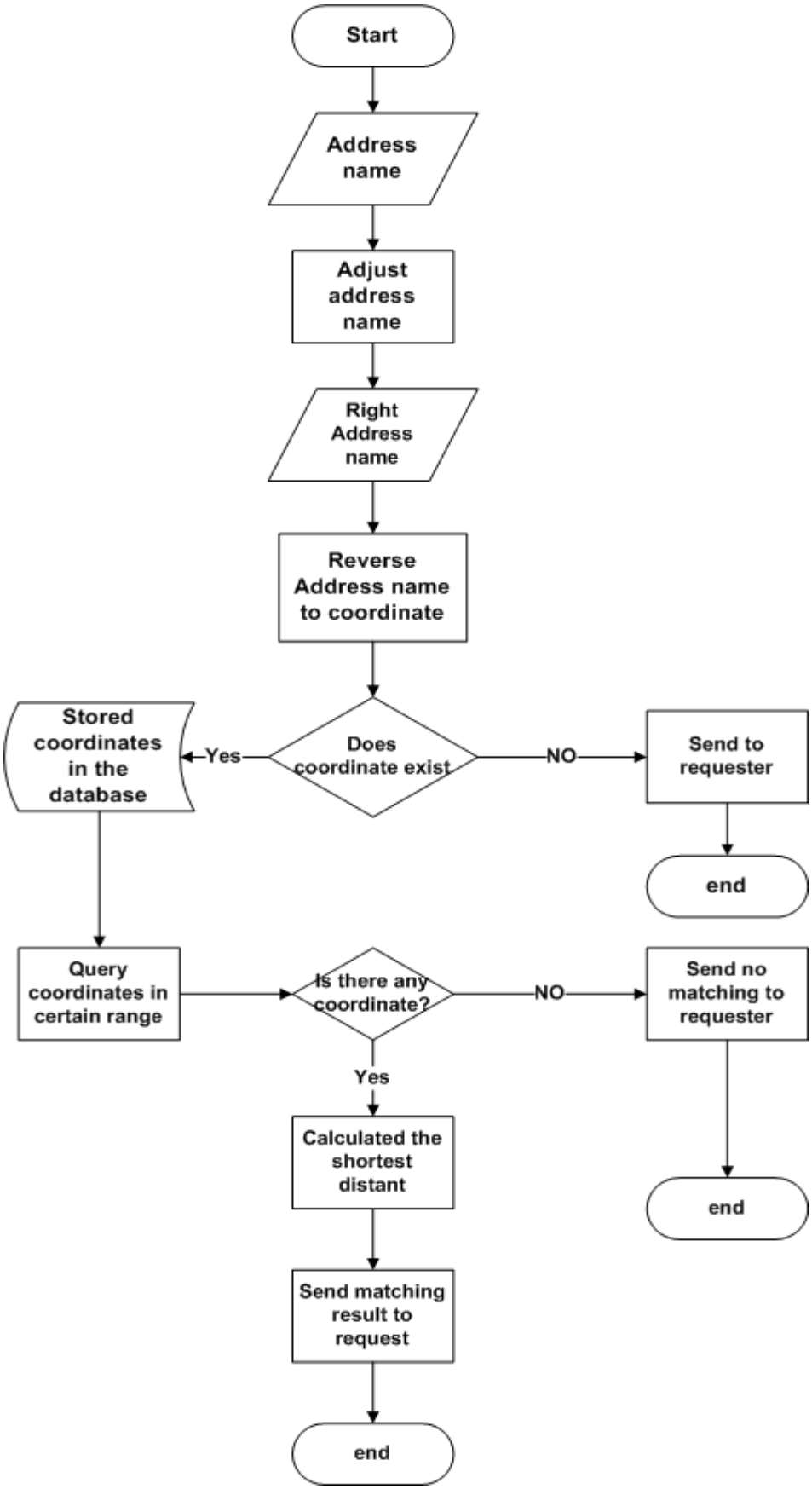

Figure 7: The flowchart of improvement of Taxishare system matching logic process

In the client side application, Android Taxishare client application shall bind with GPS feature to locate the current position of users rather than marking the location manually on the map. If Taxishare client application cannot find the start location, the default location shall be airport. The mobile number validation feature can be implemented with Taxishare client application. For iterative and incremental development, all the requirement analysis is rigid in each of iteration during the rapid development lifecycle. The requirement and design does not well document. The documentation of Taxishare system shall be produced after the project is completed for further development.

4. Conclusion

The aim of the project is to deliver Taxishare client application for Taxishare system within limit schedule and limit resource. The Taxishare client application project development demonstrates that iterative and incremental development lifecycle is an appropriate approach for a small and rapid project with self-organise development team to achieve customers' objective on schedule. The project development starts from advance preparation including risk estimation and environment set up before project iterative development. Each of development iteration focuses on resolve the project issue or produce new portion of the Taxishare client application. In addition, through the entire development lifecycle, the defect of Taxishare system matching algorithm is discovered. The report presents multiple design choice on android Taxishare client application. The software engineer and project management technique help to analysis multiple design choices and make a decision to achieve the final project goal successfully. The report also proposes the further improvement for Taxishare system and Taxishare client application on android platform. Eventually, the project was completed on schedule under the expectation.

5. Acknowledgement

I would like to express my deepest thanks to my supervisor, Dr. Ken Taylor, who offer me guidance during the project development. Thanks and appreciation to Herman Jaya who support and assist me on Taxishare system development. It is a pleasure to cooperate with CSIRO taxishare project team. I wish to express my deep sense gratitude to, Dr. Tom Gedeon, who introduces the project to me. I also would thank to, Dr. Uwe R. Zimmer, who guide me the project presentation and scientific report

writing skill. I also extend my thanks to my team member Melvin Yong, who discusses the project development with me.

6. Reference

Android API Level, Google Android Development Guide, viewed 27 May 2010, <http://developer.android.com/guide/practices/compatibility.html>

Android Compatibility, Google Android Development Guide, viewed 27 May 2010, <http://developer.android.com/guide/practices/compatibility.html>

How does SMS work?, General (GSM, SMS, Modern, WAP, GPRS etc), Logic mobile, viewed 28 May 2010, <http://www.logixmobile.com/faq/show.asp?catid=1&faqid=3>

How Does SMS work, Text Messaging Centre, viewed 25 May 2010, <http://www.tmcsms.com/how-does-sms-work.aspx>

Herman Jaya 2009, *Sprint 1.1 – Implementation*, CSIRO Taxishare, viewed 28 May 2010, <https://wiki.csiro.au/confluence/display/Taxishare/Sprint+1.1+-+Implementation >

Herman Jaya 2009, *Sprint 3 (completed)*, CSIRO Taxishare, viewed 23 May 2010, < https://wiki.csiro.au/confluence/display/Taxishare/Sprint+3+%28completed%29>

Herman Jaya 2010, *Sprint 4 (current)*, CSIRO Taxishare, viewed 24 May 2010, < https://wiki.csiro.au/confluence/display/Taxishare/Sprint+4+%28current%29>

Herman Jaya 2010, *System Algorithm*, CSIRO Taxishare, viewed 24 May 2010, < https://wiki.csiro.au/confluence/display/Taxishare/System+Algorithm>

Ian Spence and Kurt Bittner 2005, *What is iterative development*, IBM developerWorks, viewed 25 May 2010, <http://www.usq.edu.au/library/help/referencing/harvard.htm#Web_documents_and_sites>

Jennifer Hord, *How SMS Works*, How Stuff works, view 25 May 2010, <http://communication.howstuffworks.com/sms.htm>

Jim Lilley 2009, *2008 Director's Challenge Entry*, CSIRO Taxishare, viewed 23 May 2010, <https://wiki.csiro.au/confluence/display/Taxishare/Dir+Challenge+Entry>

Jim Lilley 2010, *Meeting Notes*, CSIRO Taxishare, viewed 23 May 2010, <https://wiki.csiro.au/confluence/display/Taxishare/Meeting+Notes>

Signing Your Application, Google Android Development Guide, viewed 29 May 2010, <http://developer.android.com/guide/publishing/app-signing.html>