



Bahria University
Discovering Knowledge

FINAL YEAR PROJECT REPORT

Transit Pakistan

By

Muhammad Sajjad Arif	31231	BSE
Muhammad Haris Ali	31226	BSE
Muhammad Moaz Hassan	31228	BSE

Supervised by

Engr. Wafa Qaiser

Bahria University - Karachi Campus

2016

Submission Performa

Names

- (1)..... Muhammad Sajjad Arif
(2)..... Muhamad Haris Ali
(3)..... Muhmmad Moaz Hassan

Addresses

- (1)..... H.No 11/1 Majeed SRE Karachi
(2)..... Block H-4, Falt # 18, Nazimabad, Karachi.
(3)..... H.No R-73-74 Gulshan-e-Siddique Malir City.

Transit Pakistan – Personal Transport Assistant

Project Supervisor's Name

Engr. Wafa Qaiser

This report is submitted as required for the Project in accordance with the rules laid down by the Bahria University as part of the requirements for the award of the degree of Bachelor of Engineering. I declare that the work presented in this report is my own except where due reference or acknowledgement is given to the work of others.

Signatures of students

Date

- (1).....
(2).....
(3).....

.....
.....
.....

Signature of Supervisor

Date

- (1).....

.....

Contact Information



Author(s)

Muhammad Sajjad Arif

Email : sajadarifgul@gmail.com

Phone : 03320307817

Muhammad Moaz Hassan

Email : maaz773@hotmail.com

Phone : 03332579199

Muhammad Haris Ali

Email : mharisali739@gmail.com

Phone : 03453280363

Supervisor

Engr. Wafa Qaiser Khan

Email : wafa.qaiser@bimcs.edu.pk

Phone : 03452896663



Intellectual Property Right Declaration

This is to declare that the work under the supervision of Engr. Wafa Qaiser Khan having title “Personnel Transport Assistant App” carried out in partial fulfillment of the requirements of Bachelor of Engineering in Software is the sole property of the Bahria University and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of its Authors, the University and respective supervisor. This above statement applies to all students and faculty members.

Date: 24/05/2016

Signatures of Students

- (1).....
- (2).....
- (3).....



Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of Engr. Wafa Qaiser Khan having title “Personnel Transport Assitnat App” is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: 24/05/2016

Signatures of Students

(1).....

(2).....

(3).....

Acknowledgments

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

At first, all thanks to Al-Mighty Allah who gave us the ability, strength and courage to complete the project, and then we would like to express our gratitude and appreciation towards our parents, without their help we would have not been standing here, where we are today and members of our institute Bahria University, Karachi Campus for their kind co-operation and encouragement which helped us in completion of this project.

We would like to express our deepest appreciation to all those who gave us the opportunity to complete this document and project. A special thanks to our final year project supervisor Engr. Wafa Qaiser Khan - as without her help, supervision, assistance, ideas and encouragement, this project would not have been possible.

This report could not have been written without Engr. Wafa Qaiser Khan who served us by helping and providing us with necessary guidelines not only in the development of the project but also in the preparation of this report.

Abstract

Karachi is the biggest city and financial hub of Pakistan whose population is about 14,500,000 citizens and more than 2 million people of this city commute through public transport on daily bases. There exists a vast majority of these commuters who have obscure information about public transport and their routes which led them to misguidance and wastage of time on bus stops. According to the research an average commuter waste around 20 minutes per day waiting at bus stop which is around wasting 10 days per year.

The project is develop to resolve these issues and provide a convenient platform to the commuters so that they can easily access public transport information like bus routes along with other cost-effective travel options like cab service, taxis and rickshaw. The application will let them know about the fares for the selected ride option and approximated average time to reach the destination.

During our research we did not found any currently working automated system which can address commuters this problem. In this prospective we realized the issues concerning to the commuters and came up with the solution to develop Transit Pakistan android application which can manage these uncertainties associated with Karachi's public transportation. It will also cover public transport routes, ride options, cost effective route & fares, average time for each route along with the highlighted visual representation of route on Map.

We also resolve the issue for those commuters who don't have an android phone or access to internet by providing the SMS service so that our application will empower commuters and enhance the reach of urban transport. Transport is the backbone behind the largest GDP generating city Karachi, so the problems faced by 2 million of its commuters need to be addressed. Transit will play a vital role in minimizing the problems related to Karachi transport. It will deliberately enhance the people's reliance on public transport but also increase the reach of public transport. In future after the success of transit Pakistan in Karachi we have planned to incorporate the other cities of Pakistan to our system gradually.

Keywords: Transit, Transport, Routes, Rides, Buses, Fares.

Table of Contents

TRANSIT PAKISTAN	1
ACKNOWLEDGMENTS	3
ABSTRACT.....	7
TABLE OF FIGURES.....	10
1.1. INTRODUCTION.....	12
1.1.1 PURPOSE OF THIS DOCUMENT:	12
1.1.2. OVERVIEW OF THIS DOCUMENT	12
1.2. BACKGROUND AND LITERATURE REVIEW	13
1.2.1. EXISTING SYSTEM:	13
1.2.1.1 Existing System Description:	13
1.2.1.2 Problems in Existing System:.....	13
1.3. AIM AND STATEMENT OF PROBLEM.....	14
2.1. SYSTEM ANALYSIS.....	16
2.1.1 DATA ANALYSIS	16
2.1.1.1 Data Flow Diagram.....	16
2.1.2. SYSTEM REQUIREMENTS	17
2.1.2.1. Clients, Customer and Users	17
2.1.2.2. Functional and Data Requirements	17
2.1.3. DATABASE REQUIREMENT	18
2.1.3.1. Description.....	18
2.1.3.2. Technical issues	18
2.1.3.3. Dependencies with other requirements.....	18
2.1.4. NON-FUNCTIONAL REQUIREMENTS	18
2.1.4.1. Look and feel requirements.....	18
2.1.4.2. Usability requirements.....	18
2.1.4.3. Security requirements	18
2.1.4.4. Performance requirement	18
2.1.4.5. Portability requirements	18
2.1.5. PROPOSED SOLUTIONS.....	19
2.1.5.1. User Perform Functions	19
2.1.5.2. System Perform Functions	19
2.1.6. ALTERNATIVE SOLUTIONS	19
2.2. SYSTEM ARCHITECTURE AND PROGRAM FLOW	20
2.2.1. MAJOR MODULES IN SYSTEM:	20
2.2.1.1. Transit Tool:	20
2.2.1.2. API:	20
2.2.1.3. Mobile App:	20
2.2.1.4. SMS:.....	20
2.3. DETAILED SYSTEM DESIGN.....	21

2.3.1. DETAILED COMPONENT DESCRIPTION:	21
2.3.2. <i>Component Diagram</i>	21
2.3.3. <i>UML Diagrams</i>	21
2.3.3. <i>UML diagrams</i>	21
2.4. USE CASE NARRATIVES.....	26
2.4.1. <i>Transit Tool</i>	26
3.1. DESIGN CONSTRAINTS	33
3.1.1. HARDWARE AND SOFTWARE ENVIRONMENT	33
3.1.2. END USER CHARACTERISTICS	33
3.1.3. SOFTWARE INTERFACES	33
3.2. ARCHITECTURAL STRATEGIES	33
3.2.1. ALGORITHM TO BE USED	33
3.2.1.1. <i>Permutation</i>	33
3.2.2. PROJECT MANAGEMENT STRATEGIES	34
3.2.2.1. <i>Management Activities</i>	34
3.2.2.2. <i>Project Planning</i>	35
3.2.2.3. <i>Project Scheduling</i>	36
3.2.2. DEVELOPMENT METHOD.....	37
3.2.2.1. <i>Requirements</i>	37
3.2.2.2. <i>Develop Product Backlog</i>	38
4.1. SOURCE OF DATA	40
4.2. TRANSIT API.....	40
4.2.1. <i>Locations Data</i>	40
4.2.2. <i>Routes Data</i>	40
4.2.3. <i>Vehicles Data</i>	40
4.2.4. <i>SMS Service</i>	40
4.3. TRANSIT TOOL	41
4.4. ANDROID APPLICATION	41
4.5. SMS SERVICE	41
5.1. VALIDATION.....	43
5.1.1. BLACK BOX TESTING	43
5.1.2. WHITE BOX TESTING	44
5.1.2.1. <i>Route Generations Logic</i>	44
5.1.2.2. <i>Flow Chart Route Generations Logic</i>	45
5.1.2.3. <i>Graph Notation of Route Generations Logic</i>	46
5.1.2.4. <i>Cyclomatic Complexity of Route Generations Logic</i>	47
5.1.2.5. <i>Predictive Paths of Route Generations Logic</i>	47
6.1. RESULTS	49
6.2. CONCLUSIONS	52
6.3. FUTURE WORK.....	53
7.1. CODE.....	55
7.2. REFERENCES.....	67

Table of Figures

Figure 1 - Data Flow Diagram	16
Figure 2 - Level-1 Data Flow Diagram of SMS Users	16
Figure 3 - Level-1 Data Flow Diagram of Mobile Users.....	17
Figure 4 - Component Diagram	21
Figure 5 - Actor Usecase Diagram.....	22
Figure 6 - State Machine Diagram.....	22
Figure 7 - Activity Diagram.....	23
Figure 8 - Sequence Diagram	24
Figure 9 - Context Diagram	24
Figure 10 : Architecture Diagram	25
Figure 11 - ERD Diagram.....	25
Figure 12 - Workbreak down structure	35
Figure 13 : Gantt Chart	35
Figure 14 : Project Scheduling Planning	36
Figure 15 : Spiral Model of Transit Pakistan Development	37
Figure 16 – Flow Chart Diagram	45
Figure 17 : Graph Notation of Route Generation Logic	46
Figure 18 - Main Activity View.....	49
Figure 19 - Detail Activity View	49
Figure 20 - Sample Response from SMS Service.....	50
Figure 21 - Transit Tool - Main XML View.....	50
Figure 22 - Routes Generation Window	51
Figure 23 - Transit SMS Window.....	51

CHAPTER 1 – INTRODUCTION

1.1. Introduction

The purpose of the project is to develop an application which can facilitate the Commuters who uses Public Transport and let them aware about transport routes and their options like buses, taxis, Rickshaw there average time that they require to reach the destination and estimated fares. The main objective is to enable people's access to public transport information, empowering commuters to know about other options and increase the reach of urban transport.

1.1.1 Purpose of this document:

The purpose of this document is to fully describes complete requirements related to the system that include purpose of the project, description about existing systems and the problems in the existing system.

Document also discovering, the complete data, system, functional & non-functional requirements of the system, & the proposed solutions of the project.

THIS DOCUMENT COVERING THE COMPLETE DESIGN CONSTRAINTS RELATED TO PROJECTS, HARDWARE & SOFTWARE ENVIRONMENT, IN THIS YOU ALSO HAVE TO DEFINE THE END USER CHARACTERISTICS, ARCHITECTURAL STRATEGIES, ALGORITHM TO BE USED, PROJECT MANAGEMENT STRATEGIES, DEVELOPMENT METHOD & FUTURE ENHANCEMENTS/PLANS.

DOCUMENT ALSO DEFINING THE COMPLETE SYSTEM DESIGN OF THE SYSTEMS. THIS IS ALSO COVERING COMPLETE ARCHITECTURE & PROGRAM FLOW, MODULES & SUB MODULES, & DESCRIPTION OF THE SYSTEM COMPONENTS.

In the end of the document, cover complete implementation & testing like black box testing, white box test cases.

1.1.2. Overview of this document

This document also describe the each functionality of the project Transit Pakistan which is an Android App. Define the data flow and the system requirements of the project , It give you the information about each module like Transit tool ,API, Mobile App and SMS service. It also provides the proposed solution and alternative solution of the Transit Pakistan and defines the design constraint and nonfunctional requirements of the projects.

1.2. Background and Literature Review

1.2.1. Existing System:

1.2.1.1 Existing System Description:

During our research we could not find any automated system which exists today. Commuters verbally ask from others on different bus stops because of the obscure information to let them aware about the buses to reach the desired destination.

1.2.1.2 Problems in Existing System:

In existing system there is a vital problem that commuters are not aware about the buses and their routes for that purpose they asked with each other so there is a huge probability of misguidance and wastage of time. The second problem is the fare issue commuters some time not aware about the fare to reach the destination because of which conflict arises between conductor and commuters.

1.3. Aim and Statement of Problem

Commuters of Karachi are facing transport problems on regular basis. Our research notified us of some problems that can be solved by Transit. These problems are: Obscure Information, Waiting for Selective Buses, Misguidance, and Lack of Transport Apps. Transit Pakistan is being developed to solve these uncertainties associated with Karachi's public transportation. Transit will enable people's access to public transport information like bus routes along with other cost-effective travel options like cab service. The project will cover public transport routes, ride options, cost effective route & ride option, average time for each route along with a responsive highlighted visual representation of route on Map.

Chapter 2 - System Analysis & Design

2.1. System Analysis

System Analysis is done when we develop new system and we detailed study on existing system for leading the new system. In analysis phase we emphasize on how to create the different combinations between the routes and alternative busses to reach the destination in case there is not any bus which directly lead to the desire destination.

2.1.1 Data Analysis

Data analyses is a procedure of inspection, cleaning, converting, and displaying detail with the objective of discovering useful information, recommending conclusions, and assistant for making final decision. We gathered all the related information through a reliable source in KMC (Karachi Metropolitan Corporation) department. The basic concerns is about the authentication of Buses Routes and their fairs. According to the data gathered we develop the combination between routes and visualize the different alternative routes on the google map to reach the destination.

2.1.1.1 Data Flow Diagram

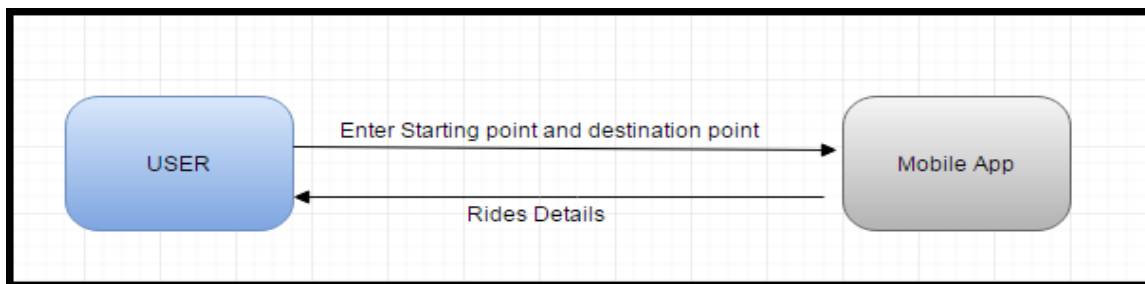


Figure 1 - Data Flow Diagram

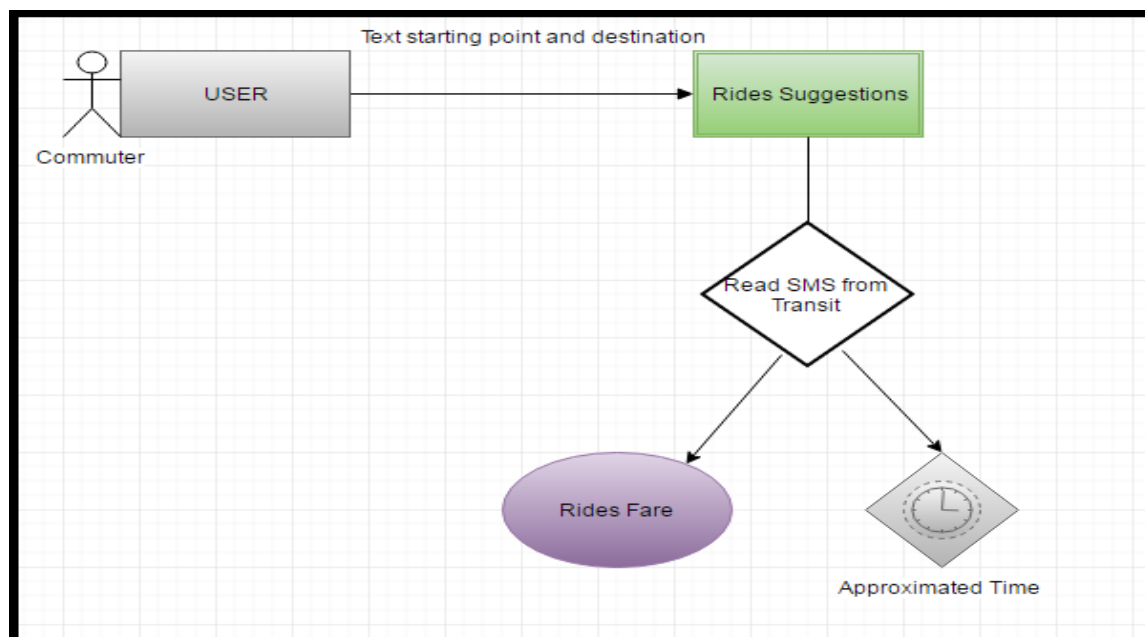


Figure 2 - Level-1 Data Flow Diagram of SMS Users

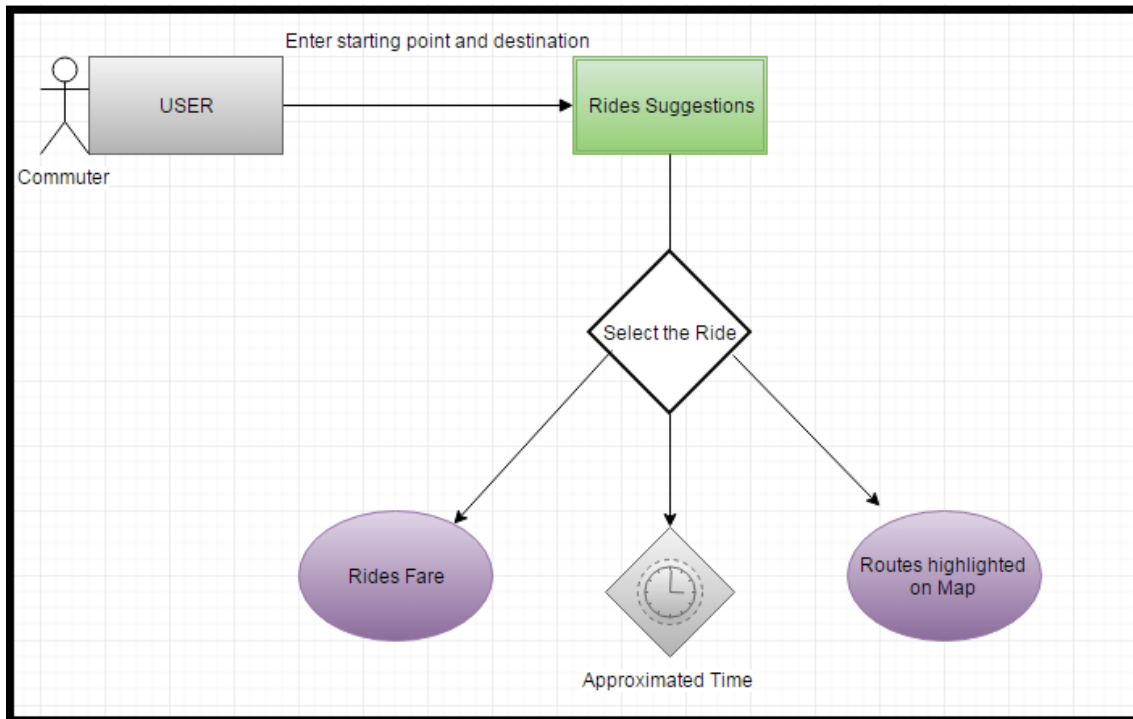


Figure 3 - Level-1 Data Flow Diagram of Mobile Users

2.1.2. System Requirements

2.1.2.1. Clients, Customer and Users

Client, Customer and User involvement in requirement gathering plays a vital role because through user involvement we can find out what the actual users need and wants. Our users can be any one he just downloaded our app from the play store and can use it for their desire purpose. Initially he inserts the starting point and destination it will simultaneously provide the transport information.

2.1.2.2. Functional and Data Requirements

Ride Information: User can view the desire buses data by inserting the initial position and destination. The application will also display the alternative rides like rickshaw and taxis with their fairs.

Routes Combination: Identifying the alternative routes for buses in case there is a bus which does not directly reach the desire destination our application will also provide the information of alternative buses which pass from that route to the destination.

Fares And Time: The Application will display the fare and time according to the user provided starting point and destination it will deliberately display the fare details as the ride selected like (bus taxi rickshaw) by user. The fares are displayed according to the Bus stops and whole ride .The approximate time is also presented so that user can be aware and differentiate between rides.

SMS: In case commuter does not have android mobile or Internet for this problem SMS service is the right solution provided to the users. The user simply text his starting point and destination in response to it server will dispatch the rides information in correspondence to the users SMS.

2.1.3. Database Requirement

2.1.3.1. Description

Transit Tool is the main tool that is used to create the backend structure. Transport information is feed into this tool which in turn formats this information to XML markup & file based database. These XML markup files are incorporated into API on transipt.pk's online information processing server which is located on cloud. This server comprises of combination of MySQL Database & File Based Database. It will provide API access to database via PHP.

2.1.3.2. Technical issues

The database must have sufficient storage capacity because of the immense number of rides detail and their routes information so that the server can deal the number of requests at the same time which can make it efficient.

2.1.3.3. Dependencies with other requirements

Transit Pakistan App have transit tool which is managed by the administrator who can perform updating and deletion of data according to the rides and routes information.

2.1.4. Non-Functional Requirements

2.1.4.1. Look and feel requirements

The transit applications user interface is designed in a certain way that it can be appealing specially making sure that it can simultaneously easily interact in an effective way without any hurdle or discrepancies so that user can be satisfied.

2.1.4.2. Usability requirements

Commuters can easily interact with application in correspondence to the location and destination selected. The rides information is displayed with their approximated time and fares. Commuters selected the ride according to their desire a map will be displayed so that they can be aware of the routes.

2.1.4.3. Security requirements

Our users are diverse they can be any one who can use our application without any restriction. All the management of data is done through transit tool which is manage by an authorized administrator.

2.1.4.4. Performance requirement

Initially we did not face any performance issues as we have covered sufficient areas of Karachi only, in future if we expend it to other cities, performance can be the issue for that purpose extra measures can be taken to resolve it.

2.1.4.5. Portability requirements

Transit Pakistan can run on any android mobile by simply download from the play store and use it. For SMS service user can text his detail to certain number like 3301 in response to it rides information is provided.

2.1.5. Proposed Solutions

The purpose of transit Pakistan App is to facilitate the commuters to access the authentic Information related to rides (fares and approximated time) so that there can be no wastage of time and misguidance. It will also empower commuters and increase the reach of urban transport.

For this we have proposed a solution by developing the transit tool and make the whole data into a usable information format so that the buses routes data can be properly managed and alternative routes combination can be created for buses in case there is a bus which does not directly reach the desire destination our application will also provide the information of alternative buses which pass from that route to the destination. For SMS users the rides information will be dispatched by text on a certain number provided by transit Pakistan to them from server.

2.1.5.1. User Perform Functions

- User can provide the initial position and destination to view the rides details.
- User can view details about the rides options like buses rickshaw taxis.
- User can view the approximated fair and time.
- User can view the routes from google map.
- Incase users doesn't have android phone or internet access he can simply make SMS to 3310 service provided in response to it rides details will be send to him.

2.1.5.2. System Perform Functions

The user provided the position and destination the, App will send the query to online server at Transit.pk via its API. The Information processing server at Transit.pk will process the information & apply its algorithms. The results of these processes will be sent back to user on his/her smartphone.

2.1.6. Alternative Solutions

Alternative routes for buses is displayed in case there is a bus which does not directly reach the desire destination our application will also provide the information of alternative buses which pass from that route to the destination.

2.2. System Architecture and Program Flow

2.2.1. Major Modules in System:

- Transit tool
- API
- Mobile app
- SMS

2.2.1.1. Transit Tool:

Transit Tool is the main tool that is used to create the backend structure. Data collected from KMC is utilized in a certain way to easily manage and make this Collected Data into usable information format. Transport information is feed into tool which formats this information to XML markup & file based database. Transit Tool is developed on Visual Studio format using C# Programming language. Admin have the rights to make changes according to the requirement he can update or delete the information as needed.

2.2.1.2. API:

The development of API is the main important module which contains all the information related to routes buses and fares. We used PHP to develop API the Information Processing Server of Transit then accessed via API which processes queries & returns results in XML object format.

2.2.1.3. Mobile App:

Transit App is a mobile application developed for Android Smartphones. Users will enter their Starting Point & Destination Point details in the app which will send the query to online server at Transit.pk via its API. The Information processing server at Transit.pk will process the information & apply its algorithms. The results of these processes will be sent back to user on his/her smartphone. Through the app commuters can easily aware about the information related to rides, fares and approximated time to reach the destination. This app will also display the map which shows the highlighted routes and alternative routes to make them aware so they can select the best among them.

2.2.1.4. SMS:

If the commuters don't have access to any smartphone or internet, they will have the option to use Transit's SMS service. Commuters will simply send a message containing their Starting Point & Ending Point to Transit's SMS service number. This service will then process the message & will send back the suggestions in plain text format.

2.3. Detailed System Design

2.3.1. Detailed Component Description:

The transit tool manage the data collected from KMC in a form of XMLs containing the rides information, bus stops, fares starting point and destination. The tool also makes combination between routes so that user can be aware about the alternative buses to reach the destination. All the manipulation of data is handled from transit tool admin can update it according to their requirement.

API is the most important module which contain the information related to the transport by accessing the information processing server .The API processes queries & returns results in XML object format.

Mobile app is designed to let the commuters access the transport information by simply entering the starting point and destination. It will eventually display the rides options depends on the commuters to select the best option among them, according to the commuters selected ride it will display a map which demonstrate the whole route to make them aware, Fares and approximated time.

SMS service is introduced for those commuters who don't have access to any smartphone; they will have the option to use Transit's SMS service. Commuters will simply send a message containing their Starting Point & Ending Point to Transit's SMS service number. This service will then process the message & will send back the suggestions in plain text format.

2.3.2. Component Diagram

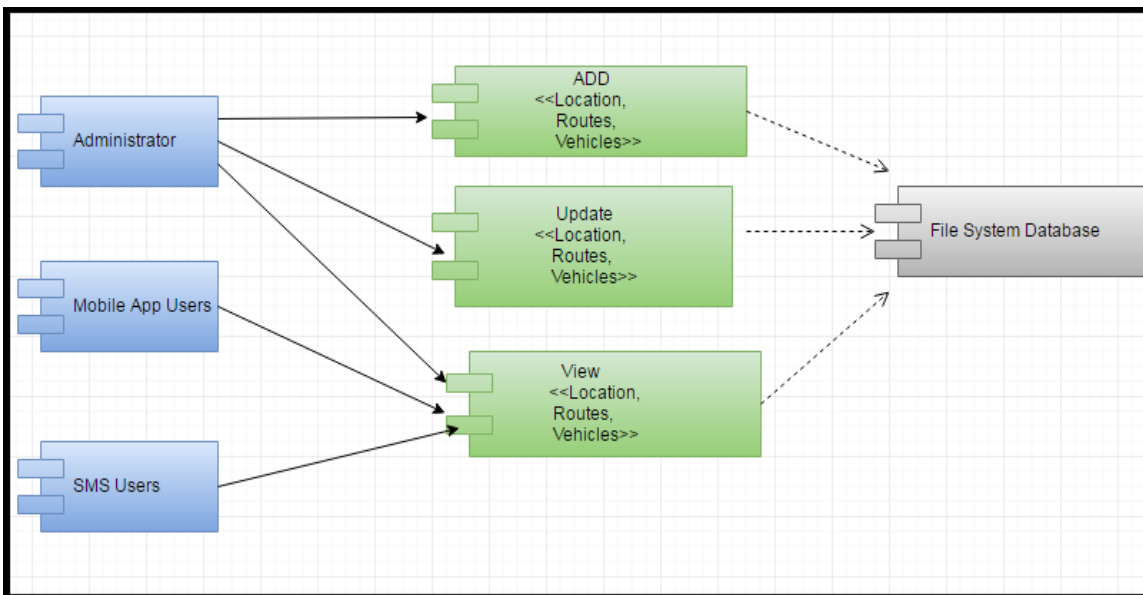


Figure 4 - Component Diagram

2.3.3. UML diagrams

UML diagrams are used to describe the system architecture. UML 2.0 specification provides thirteen kinds of diagrams. We suggest these diagrams: Use case, Class, Data

Flow, Sequence, Context, Component, and Collaboration diagrams. This is motivated by the need to define a static representation of the code and its runtime behaviour. Listed diagrams are sufficient to fulfil this purpose. To create UML diagrams, we use MS Visio tool.

2.3.3.1. Actor Use Case Diagram

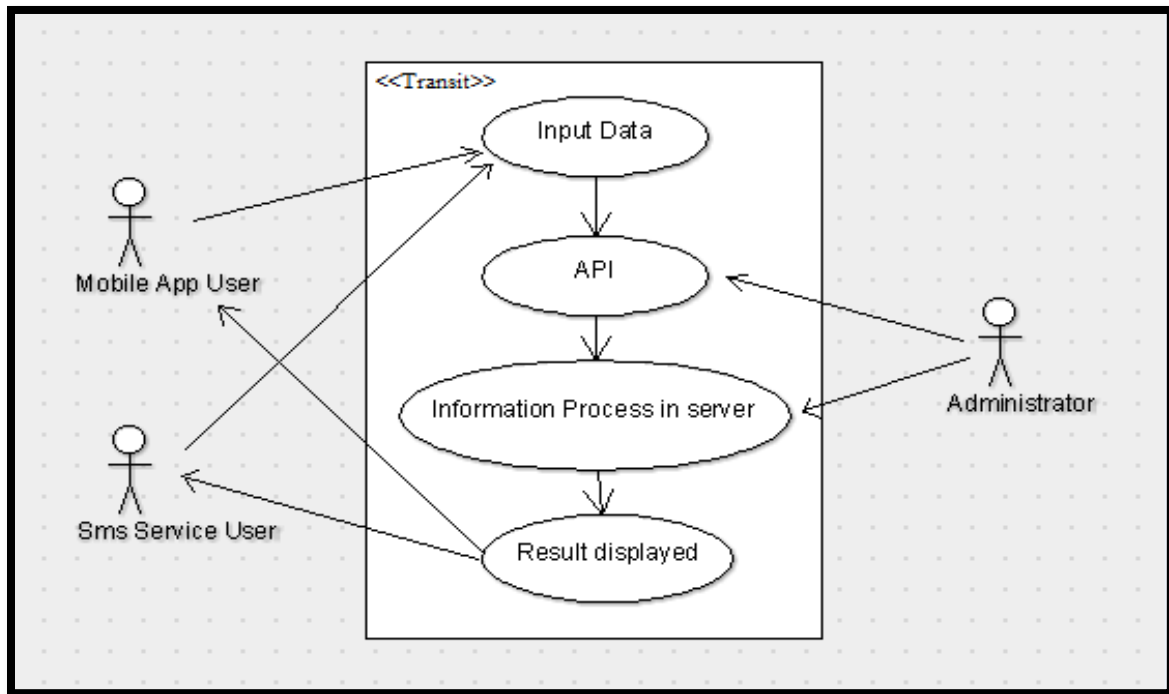


Figure 5 - Actor Usecase Diagram

2.3.3.2. State Machine Diagram

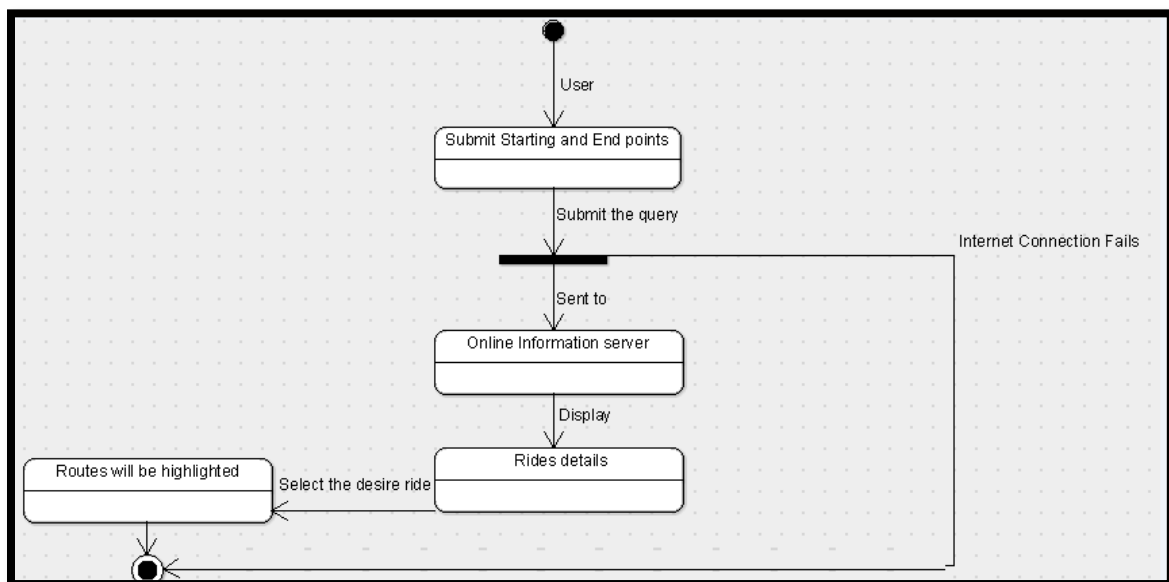


Figure 6 - State Machine Diagram

2.3.3.3. Activity Diagram

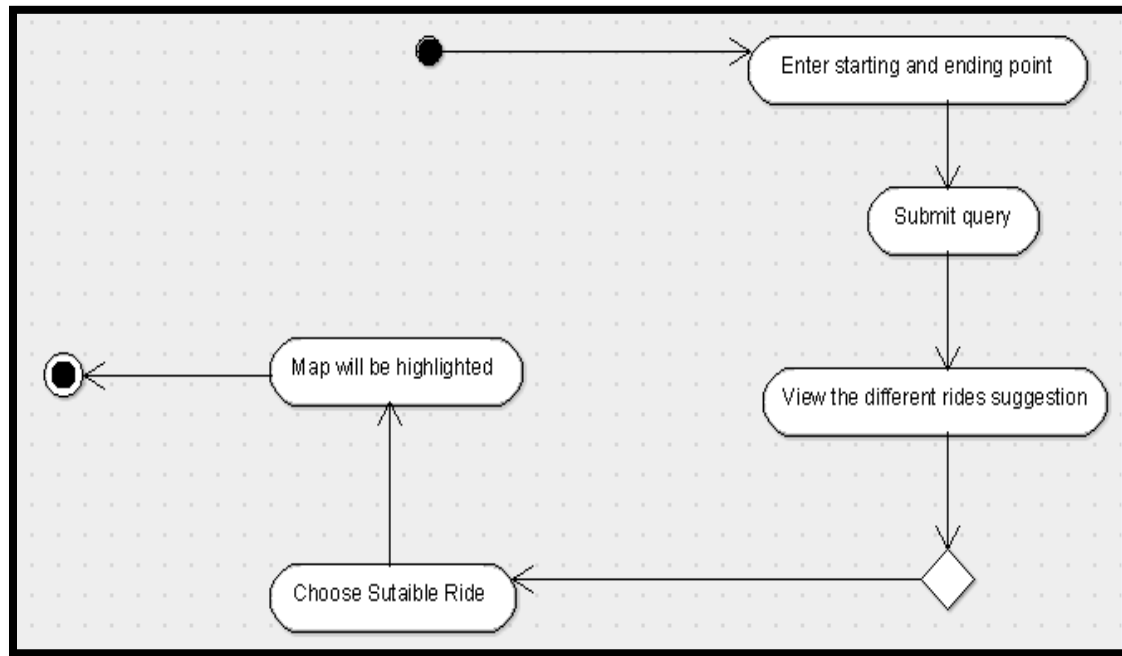


Figure 7 - Activity Diagram

2.3.3.4. Sequence Diagram

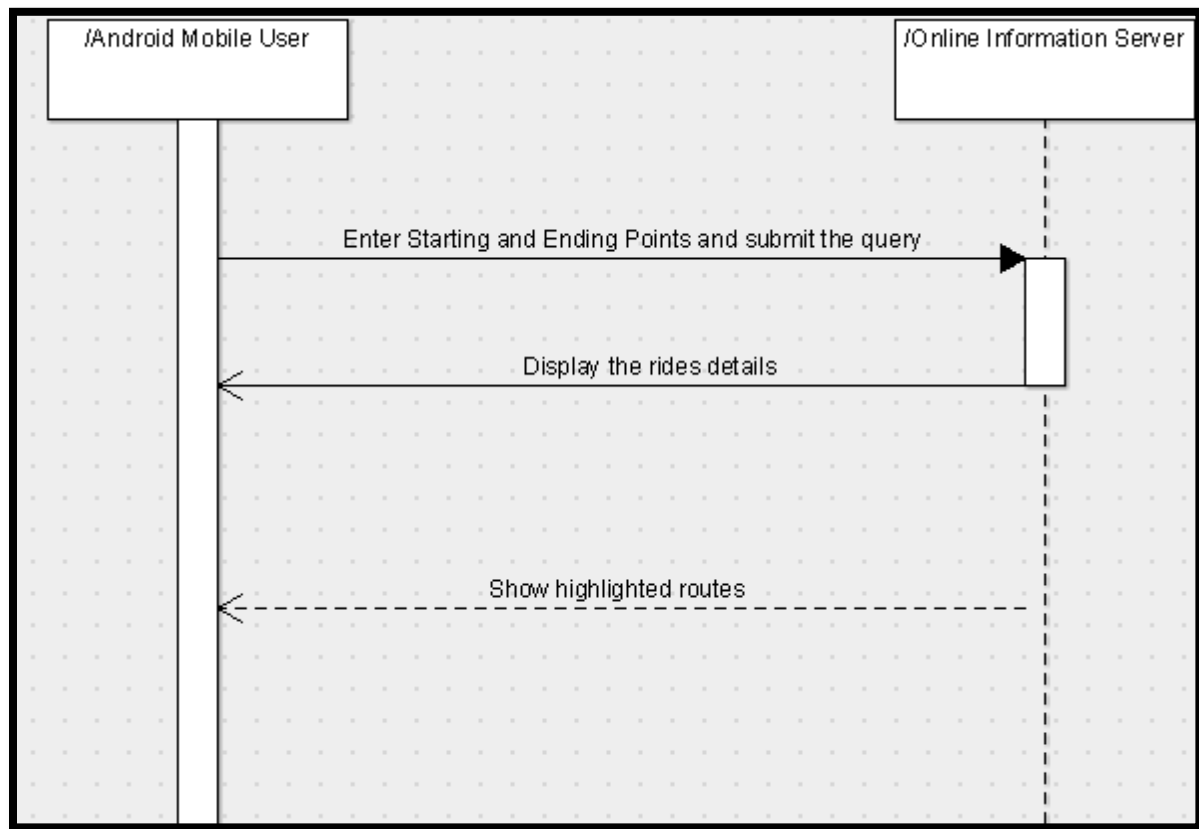


Figure 8 - Sequence Diagram

2.3.3.5. Context Diagram

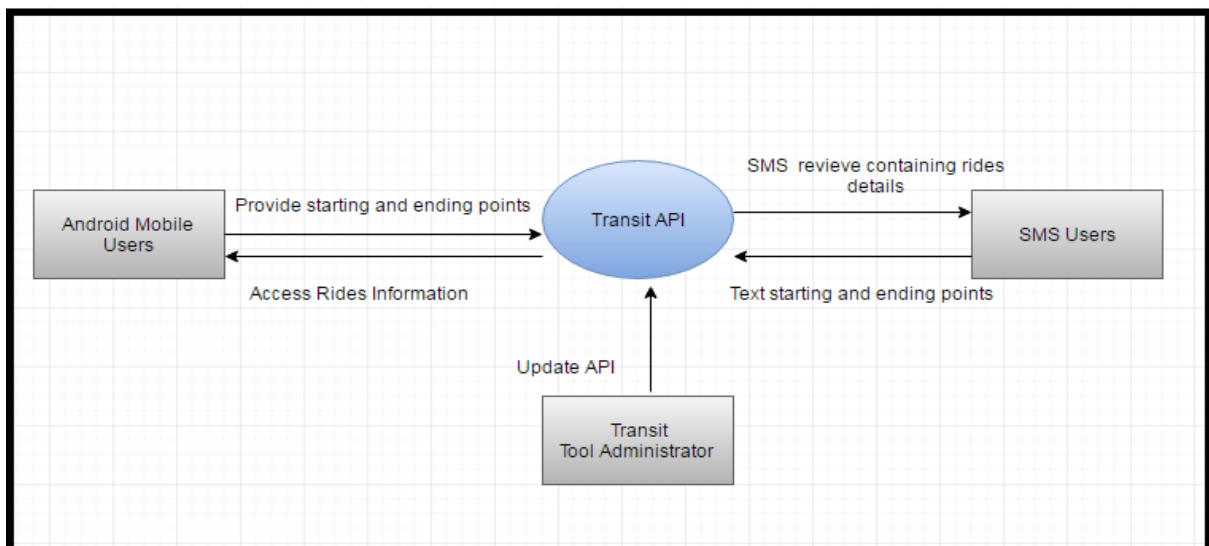


Figure 9 - Context Diagram

2.3.3.6. Architecture Diagram

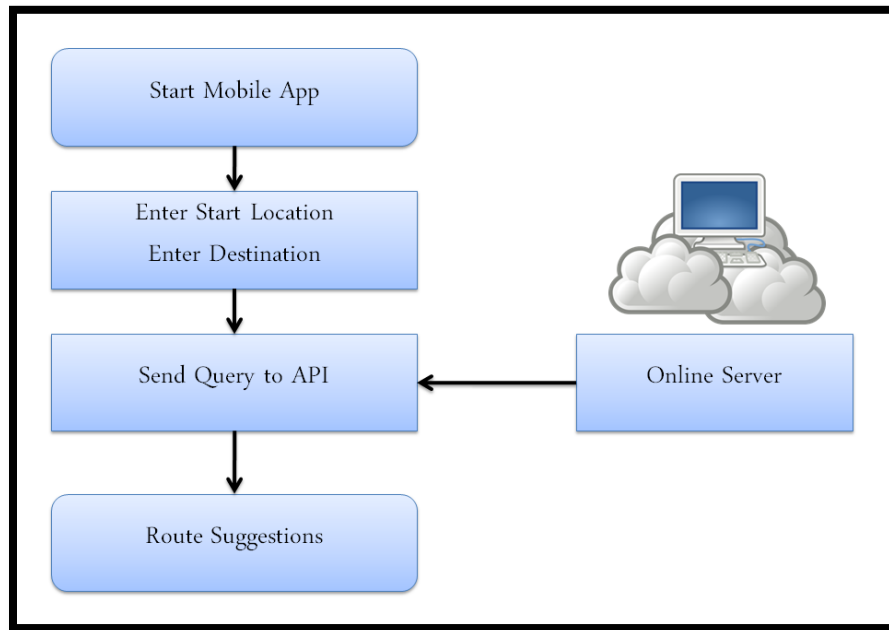


Figure 10 : Architecture Diagram

2.3.3.7. ERD Diagram

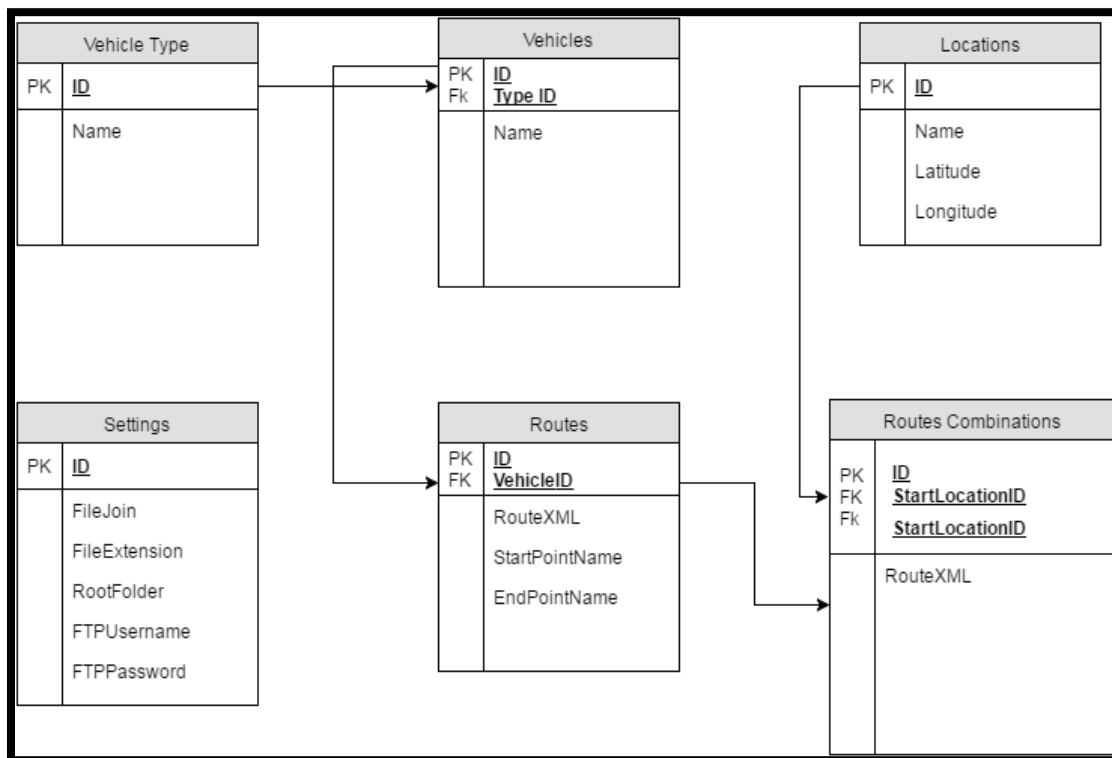


Figure 11 - ERD Diagram

2.4. Use Case Narratives

User, Admin, KMC

Use Case ID	Primary Actor	Use Cases
UC-1	Admin	Add Vehicle Route Data
UC-2	Admin	Vehicle
UC-3	Admin	Locations
UC-4	Admin	Settings
UC-5	User (Commuters)	Mobile app
UC-6	User (Commuters)	SMS

2.4.1. Transit Tool

2.4.1.1. Usecase Add Vehicles Route Data

Use Case ID:	UC-1		
Use Case	Add Vehicle Route Data		
Created By:	Muhammad Moaz	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	Admin		
Description:	The administrator can add the information related to the rides option their fare, bus stop and routes combination.		
Trigger:	The administrator has to build the tool		
Preconditions:	The User must be administrator who have the rights to manage the tool.		
Post conditions:	The XMLs file are created		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on add vehicle routes data. 2. Select vehicle type. 3. Select vehicle name. 4. Add the route fares. 5. Add Starting point of the bus stop. 6. Add next bus stop according to the bus routes. 7. Add end point of the bus 8. Formate XML 9. Generate Vehicle Route XML 10. Generate Routes XML. 		

	11. Click Routes 12. Click on Start Creating Routes Combination Files
Alternative Flows:	No alternative Flow in the process
Exceptions:	<ol style="list-style-type: none"> 1. In step 2 vehicle must be selected to get the vehicles name. 2. Select the starting point to get the latitude and longitude 3. Select the next busstop to get the latitude and longitude. 4. Select the end point to get the latitude and longitude. 5. Formate XML to set the tags in a proper formate. 6. Generate vehicle XML to create the file. 7. Generate routes XML to create the routes combination.
Includes:	None
Frequency of Use:	The admin use it frequently to manage the ride information according to the routes and fare updation.
Special Requirements:	For this use case the collection of transport data is required.
Assumptions:	The proper XML containing the ride and routes information is formed.
Notes and Issues:	No issues.

2.4.1.2. Vehicles Usecase

Use Case ID:	UC-2		
Use Case Name:	Vehicle		
Created By:	Muhammad Moaz Hassan	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	Admin		
Description:	The administrator can add new vehicle type and delete existing vehicle.		
Trigger:	The administrator has to build the tool.		
Preconditions:	The User must be administrator who have the rights to manage the tool.The data of transport detail is must.		
Post conditions:	The vehicle type and name is added or deleted		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on the vehicle option. 2. Add vehicle type. 3. Add vehicle name. 4. Delete vehicle type. 5. Delete vehicle name. 		

Alternative Flows:	No alternative Flow in the process
Exceptions:	In step 2 if vehicle type is not added it will not be displayed on add vehicle route data vehicle type and name.
Includes:	None
Frequency of Use:	The admin use it frequently to insert vehicle type and name.
Special Requirements:	For this use case the collection of transport data is required.
Assumptions:	The vehicle type and name will be added in XML.
Notes and Issues:	No issues.

2.4.1.3. Usecase Locations

Use Case ID:	UC-3		
Use Case Name:	Locations		
Created By:	Muhammad Moaz Hassan	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	Admin		
Description:	The admin can update the ride routes location.		
Trigger:	The administrator has to build the tool		
Preconditions:	He must have the data of all location.		
Post conditions:	The location can be access from add vehicle route data option starting point, next bus stop, and End point.		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on the location option. 2. Add location name. 3. Add location Latitude and longitude 4. Click on add new location 5. Update the location name and latitude and longitude. 6. Click updateselection location 7. Delete the location name and their latitude longitude. 8. Click on delete Selected location 		
Alternative Flows:	No alternative Flow in the process		
Exceptions:	1. Incase location name, latitude, longitude is not added it will not display on Add Vehicle Routes Data.		
Includes:	None		
Frequency of Use:	Admin can use it frequently after inserted location information.		

Special Requirements:	For this use case the data of the locations in the city must.
Assumptions:	The locations are displayed on Add Vehicle Routes Data so that the busstops can be added in XML.
Notes and Issues:	No issues.

2.4.1.4. Settings Usecase

Use Case ID:	UC-4		
Use Case Name:	Settings		
Created By:	Muhammad Moaz Hassan	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	Admin		
Description:	The admin can simply update the file join, file extension, root folder.		
Trigger:	Updation from settings option will directly reflected in the XML files created.		
Preconditions:	He must insert the bus stops location name so that the combination is made.		
Post conditions:	The routes combination manage		
Normal Flow:	<ol style="list-style-type: none"> 1. Insert the file join symbol. 2. Click on Update file join option. 3. Insert the file extension 4. Click on Update File Extension 5. Insert the folder path 6. Click on update root folder 		
Alternative Flows:	No alternative Flow in the process		
Exceptions:	None		
Includes:	None		
Frequency of Use:	None		
Special Requirements:	For this use case the data of the transport root is must to be inserted in Add Vehicle Routes Data		
Assumptions:	None		
Notes and Issues:	No issues.		

2.4.1.5. Mobile App Usecase

Use Case ID:	UC-5
Use Case Name:	Mobile App

Created By:	Muhammad Moaz Hassan	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	User (commuters)		
Description:	The User will insert the starting and destination points. The rides information is displayed with their approximated time and fairs. Commuters selected the ride according to their desire, a map will be displayed so that they can be aware of the routes.		
Trigger:	The user must enter the destination it will eventually display the rides details.		
Preconditions:	The internet access is must.		
Post conditions:	The ride whole route will be displayed with fare and approximated time.		
Normal Flow:	<ol style="list-style-type: none"> 1. Insert the starting point 2. Insert the destination 3. Rides detail will be displayed 4. Select the ride option 5. Ride fare ,approximate time and route on map will be displayed 		
Alternative Flows:	No alternative Flow in the process		
Exceptions:	It is must to insert the starting and destination point, if not inserted it will not display the rides detail.		
Includes:	None		
Frequency of Use:	The User can use it frequently to be aware of the rides options.		
Special Requirements:	It require the insertion of starting and destination point.		
Assumptions:	The ride detail can be attain containing the routes information, approximate time and fare.		
Notes and Issues:	No issues.		

2.4.1.6. Mobile App

Use Case ID:	UC-6		
Use Case Name:	SMS		
Created By:	Muhammad Moaz Hassan	Last Updated By:	Muhammad Haris Ali
Date Created:	21-4-2016	Last Revision Date:	21-4-2016
Actors:	User (commuters)		
Description:	If the commuters don't have access to any smartphone, they will have the option to use Transit's SMS service. Commuters will		

	simply send a message containing their Starting Point & Ending Point to Transit's SMS service number. This service will then process the message & will send back the suggestions in plain text format.
Trigger:	The user must text the starting point and destination and text it to the number provide by transit Pakistan.
Preconditions:	Mobile Balance is must for SMS.
Post conditions:	The ride details can be attain by making text to the number provide by transit Pakistan.
Normal Flow:	<ol style="list-style-type: none"> 1. Insert the starting point 2. Insert the destination 3. Text it to the number provide by transit Pakistan. 4. The rides detail will be send to him in response to the SMS.
Alternative Flows:	No alternative Flow in the process
Exceptions:	None
Includes:	None
Frequency of Use:	The User can use it frequently to be aware about the rides options.
Special Requirements:	It require the insertion of starting and destination point to send it to the number provide by transit Pakistan.
Assumptions:	The ride detail can be attain containing the routes information, approximate time and fare.
Notes and Issues:	No issues.

Chapter 3 - Design Consideration

3.1. Design Constraints

In application design, a design constraint defines some limitation on the circumstances under which application is developed, or on the requirements of the application.

3.1.1. Hardware and Software Environment

The hardware that we use in our project is USB GSM modem for send and receive SMS incase if the user do not have smart phone so he can easily use our service by feature phone.

The language we have used for the development of our transit tool that is use to create the API is C#.

- Front end: Android.
- Back end: File System Database
- XML as API Response
- PHP to process API Queries
- Information Processing Server (Domain + Hosting)

3.1.2. End User Characteristics

The end user is a person who uses this application after it has been fully developed and published. The user will provide the starting point from where he need the ride and select destination and click on submit button. After users submit the information, the query will be process by online information server and result will be displayed on second user interface which shows the details of different rides along with the respective time and average cost. The details of different rides will elaborate the information related to the transports which goes from this route. Below the details there will be a Google Map which shows the shortest and suggested routes. It depend on the user to opt required respective route and map will be highlighted respectively.

3.1.3. Software Interfaces

The main software interface of our project is Transit Pakistan's API. Other software interface such as Mobile Apps on Android and iOS platform, Web pages and SMS service will use this API to get related information form Transit Pakistan Information Processing Server.

3.2. Architectural Strategies

3.2.1. Algorithm to be used

Following algorithms are used in our application.

3.2.1.1. Permutation

We use permutation to develop different routes combinations. It is main logic of our Application.

3.2.2. Project Management Strategies

3.2.2.1. Management Activities

3.2.2.1.1. Project Overview Statement

Project Overview Statement	Project Name	Project No.	Project Manager
	Transit Pakistan		Sajjad Arif

Problem/Opportunity

Commuters of Karachi are facing transport problems on regular basis. Our research notified us of some problems that can be solved by Transit. These problems are: Obscure Information, Waiting for Selective Buses, Misguidance, Lack of Transport Apps

Goal

Transit Pakistan is being developed to solve these uncertainties associated with Karachi's public transportation.

Objective

Transit will enable people's access to public transport information like bus routes along with other cost-effective travel options like cab service. The project will cover public transport routes, ride options, cost effective route & ride option, average time for each route along with a responsive highlighted visual representation of route on Map.

Success Criteria

Transit will become success story if we will get the accurate data from authorize.

All the data must be transform into same platform.

Integration of our data to the Application.

Response from the server.

Assumptions, Risk & Obstacles

The user may enter wrong starting and point names.

The App may not be installed in user's smart phone due to the low memory.

Prepared By M.Sajjad Gull M.Haris Ali M.Moaz Hassan	Date 09 November ,2015	Approved By Engr. Wafa Qaiser Khan	Date 4 December ,2015
--	---------------------------	--	--------------------------

3.2.2.1.2. Project Costing

- Domain and Hosting Charges.
 - Domain Rs 4,400

- Hosting Rs 5,500
- GSM USB Modem
 - Rs 2,000

3.2.2.2. Project Planning

The Software's used in this application will be:

- MS Visio for UML Diagramsilms
- MS Visual Studios 2013 for Transit Tool.
- MS Excel for Gantt chart.
- MS Word for Documentation (SRS, PMP, Research Paper, Report etc).
- MS Project and Adobe Photoshop for Graphics Designing.

3.2.2.2.1. Work Break Down Structure

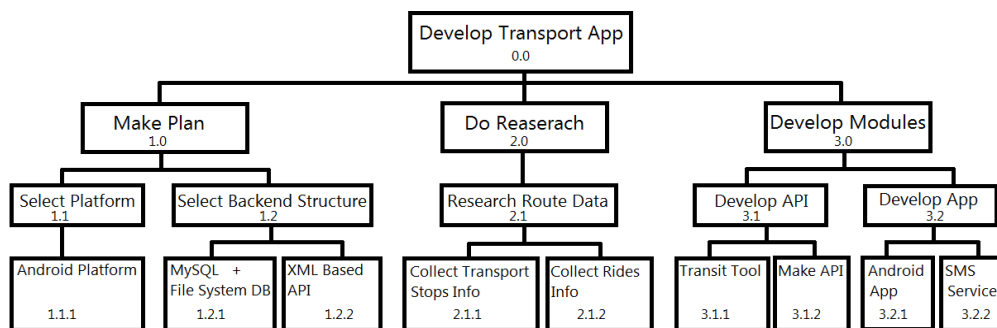


Figure 12 - Workbreak down structure

3.2.2.2.2. Gantt Chart

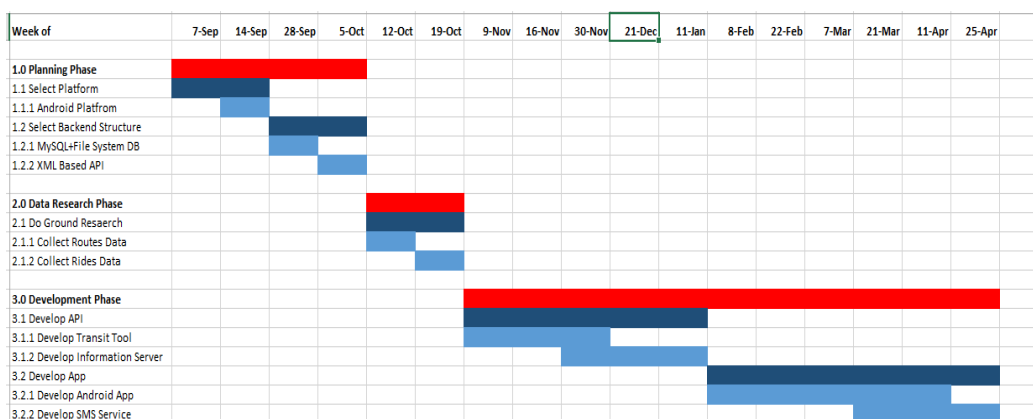


Figure 13 : Gantt Chart

3.2.2.2.3. Monitoring and reporting mechanisms

- Weekly formal meetings with Supervisor.
- Maintain Records in Meeting Logs.

- Stand up meetings with Group members.

3.2.2.3. Project Scheduling

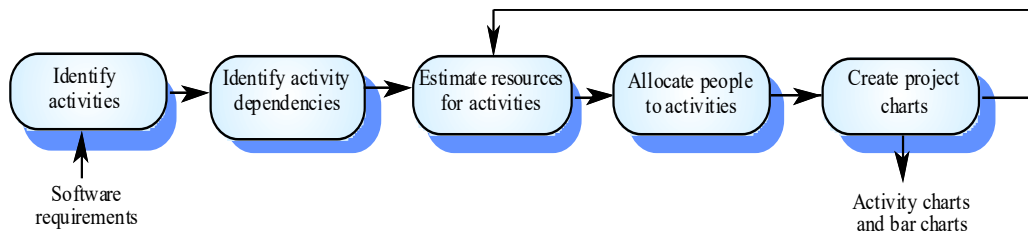


Figure 14 : Project Scheduling Planning

3.2.2.4. Risk Management

3.2.2.4.1. Risk identification

- Server
- Team risks
- Budget and cost risk
- Analysis and Design risk

3.2.2.4.2. Risk analysis

- Evaluate the moments of risk

3.2.2.4.3. Risk planning

- Making plan to remove effects of risk.
- Making contingency plans, which help us in disasters and many other risks.
- Making strategies to overcome the risk.
- We handle all the risk because we have second plan for any uncertain problem.

3.2.2.4.4. Risk monitoring

- Monitor the risks during the project.

3.2.2. Development Method

The process model used for the Transit Pakistan project is the Spiral Model.

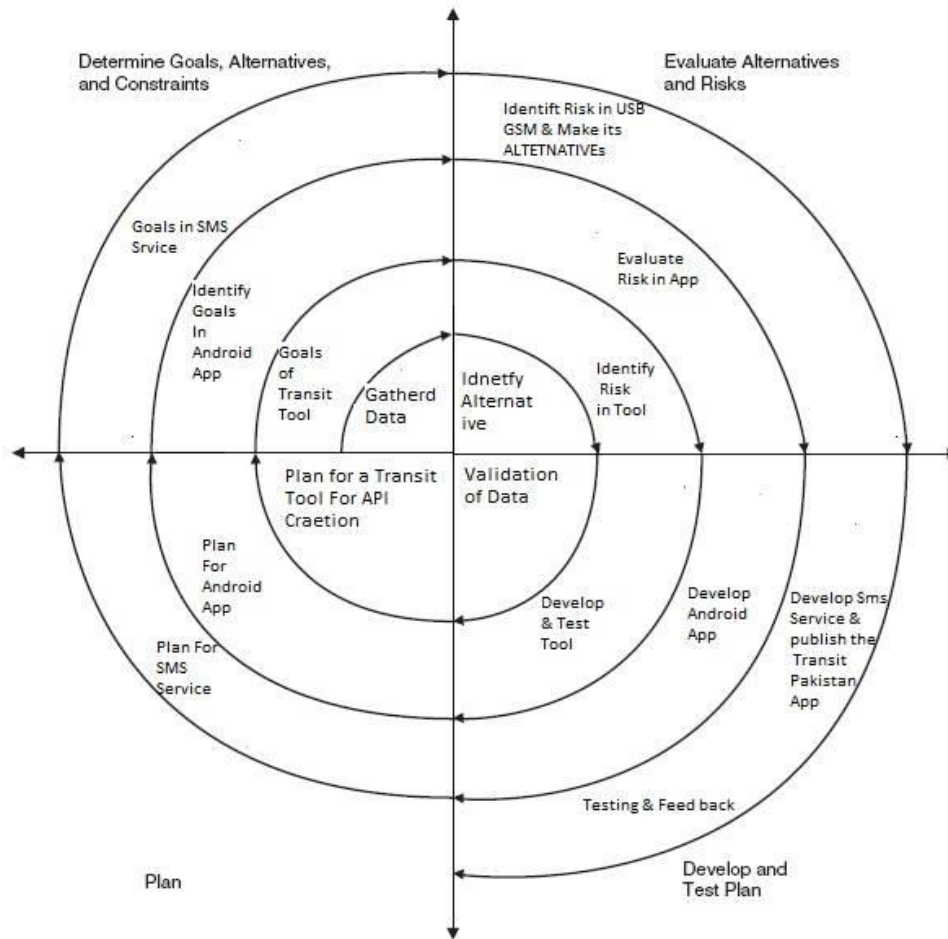


Figure 15 : Spiral Model of Transit Pakistan Development

The decision to use the Spiral methodology is due to the following characteristics of the project:

- The product definition is unstable.
- Because the cost and risk evolution is important.
- Requirement are complex.
- The buses routes might be change.
- More buses might be introduced in future.

3.2.2.1. Requirements

- We need data of buses and its routes from the KMC and Karachi Transport Authority.
- Authority. Verify all the data for the accurate results.

3.2.2.2. Develop Product Backlog

Product Backlog
1. Design application for commuters of Karachi.
2. Give them all the possible ride options.
3. Give them estimate time and cost of bus.
4. Give options of two way routes.
5. Provide SMS service for feature phone users.

Chapter 4 – IMPLEMENTATION

4.1. Source of Data

We have gathered the data from KMC and then Process that data through Transit Tool. We have successfully complete with 100+ Buses and 2,500+ different routes combinations.

4.2. Transit API

4.2.1. Locations Data

This part of our API will allow you to access the names of locations that are currently in our coverage. The following snippet shows you how to invoke locations api parts. As of May 2016, Transit Pakistan only provides coverage in Karachi, Pakistan. So the CITY attribute in your post will be 'karachi' in small alphabets because it is case sensitive. In the above response each new location is wrapped inside <LocationName> attribute. All <LocationName> attributes are then merged under <Locations> attribute which act as a document root.

4.2.2. Routes Data

This part of our API will allow you to access the route data on the given set of Start Location and Destination that are currently in our coverage. The following snippet shows you how to invoke routes api parts. As of May 2016, Transit Pakistan only provides coverage to selective 12,000 routes in Karachi, Pakistan. Location names are case sensitive so you must write the correct name. For a full list of names of location refer to part 1 - Locations data. In the above response each new vehicle is wrapped inside <Vehicle> attribute. Each <Vehicle> attribute has its own set of attribute that can be interpreted to get more information about that particular vehicle on this route. All <Vehicle> attributes are then merged under <Route> attribute which act as a document root.

4.2.3. Vehicles Data

This part of our API will allow you to access the vehicles data that are currently in our coverage. The following snippet shows you how to invoke vehicles api parts. This part of the API is currently unstable and should not be used in production. As of May 2016, Transit Pakistan only provides coverage to selective 112+ contracts of vehicles in Karachi, Pakistan. In the above response vehicle data is wrapped inside <Vehicle> attribute. Each <Vehicle> attribute has its own set of attribute that can be interpreted to get more information about that particular vehicle.

4.2.4. SMS Service

This part of our API will allow you to access the same data via SMS service that you can access via our online API. The following snippet shows you how to invoke SMS api

parts. This part of API only supports route data. As of May 2016, Transit Pakistan only provides coverage to selective 15000+ routes combinations in Karachi, Pakistan.

4.3. Transit Tool

For the development of our Transit Tool we have use C# and SQL Server and the main logic we have use for the different routes combination is Permutation.

4.4. Android Application

For the development of android application we have use eclipse and file system data base for the response of API.

4.5. SMS Service

For the SMS service we have use USB GSM modem Huawei E 169 G and our Custom software that will send and receive SMS as per our requirements.

Chapter 5 – Testing

5.1. Validation

5.1.1. Black Box Testing

Black box is the technique of software testing that examines the functionalities of the products without expressive its internal structure. Major focus in black box testing is on the functionalities of the product not on the internal structure and source code.

S.No	Test Condition	Expected Outcome	Output
1	Does the application use a consistent layout on every activity?	Consistent Layout across all activity	Pass
2	Does the main activity clearly identify the application's purpose?	A concise purpose statement on the main activity	Pass
3	Does the project use aware names and terms?	Easy to understand terms are used	Pass
4	Does your project use consistent navigation?	A consistent navigation across all activity.	Pass
6	Do all activates on the app contain navigational options?	No Navigational option on splash activity.	Fail
7	Does every activity have a meaningful title?	Concise and meaningful titles on each activity.	Pass
8	Do your activity have a consistently display?	Activities are display quickly.	Pass
9	Are all the text and labels on your each activity sequentially arranged?	Consistent layout for controls within an activity.	Pass
10	Are all controls aligned?	Aligned Controls	Pass
11	Do the controls work correctly?	Controls function properly	Pass
12	Have you used the correct and consistent color for background?	Correct and consistent colors for general screen background	Pass
13	Is the color used for field consistent and suitable?	Suitable and decent field color .	Pass
14	Is consistency being maintained while using colors?	Consistency in using colors	Pass
15	If App will not response is there any alternative?	We provide SMS service as well.	Pass
16	Is there privacy concerned in your app?	We never disclose the current location of user.	Pass
17	Easy to use actions?	All the actions easy to remember.	Pass
18	Too many actions?	No usage of too many actions we just have 3 activity.	Pass
19	Is there any appropriate error messages being displayed to user?	Appropriate Error messages are displayed in case of server down and incorrect input.	Pass

20	Is there any special icon?	Icon available to show estimated cost & time of bus.	Pass
----	----------------------------	--	------

5.1.2. White Box Testing

5.1.2.1. Route Generations Logic

```
private ArrayList _GenerateRoutes(string[] XMLArray, string VehicleType, string
VehicleName, string StopFair, bool TwoWay)
{
    (1)ArrayList BusStopsNodes = new ArrayList();

    for ( (2)int i = 6; (3)i < XMLArray.Length - 4; (6)i = i + 5)
    {
        (4) string NodeXML = XMLArray[i] + "\n"
            + XMLArray[i + 1] + "\n"
            + XMLArray[i + 2] + "\n"
            + XMLArray[i + 3] + "\n"
            + XMLArray[i + 4] + "\n";

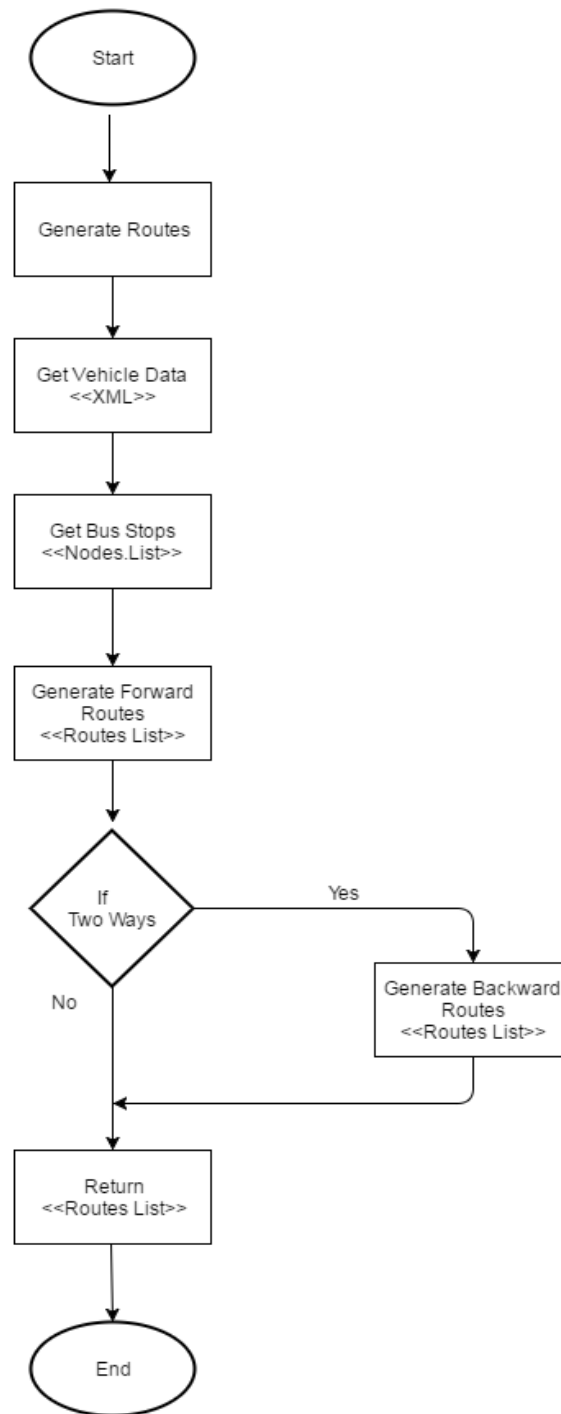
        (5)BusStopsNodes.Add(NodeXML);
    }

    (7)ArrayList ListOfRoutes = new ArrayList();

    (8)ListOfRoutes.AddRange(_GenerateForwardRoutesXML(BusStopsNodes,
VehicleType, VehicleName, StopFair));

    (9) if (TwoWay)
    {
        (10)
        ListOfRoutes.AddRange(_GenerateBackwardRoutesXML(BusStopsNodes, VehicleType,
VehicleName, StopFair));
    }

    (11)return ListOfRoutes;
}
```

5.1.2.2. Flow Chart Route Generations Logic**Figure 16 – Flow Chart Diagram**

5.1.2.3. Graph Notation of Route Generations Logic



Figure 17 : Graph Notation of Route Generation Logic

5.1.2.4. Cyclomatic Complexity of Route Generations Logic

Number of Node (N) = 11

Number of Edges (E) = 12

Region = E-N+2

= 12—11+2

= 3

Predicate = Region+1

=3+1

=4

5.1.2.5. Predictive Paths of Route Generations Logic

Path1 = 1,2,3,4,5,6,7,8,9,10,11

Path2 = 1,2,3,4,5,6,3,7,8,9,10,11

Path3 = 1, 2,3,4,5,6,7,8,9,11

Path4 = 1,2,3,4,5,6,3,7,8,9,11

Chapter 6 – Results & Conclusion

6.1. Results

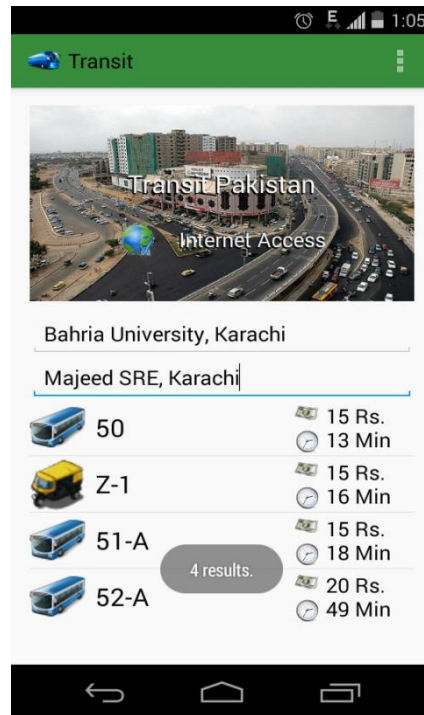


Figure 18 - Main Activity View

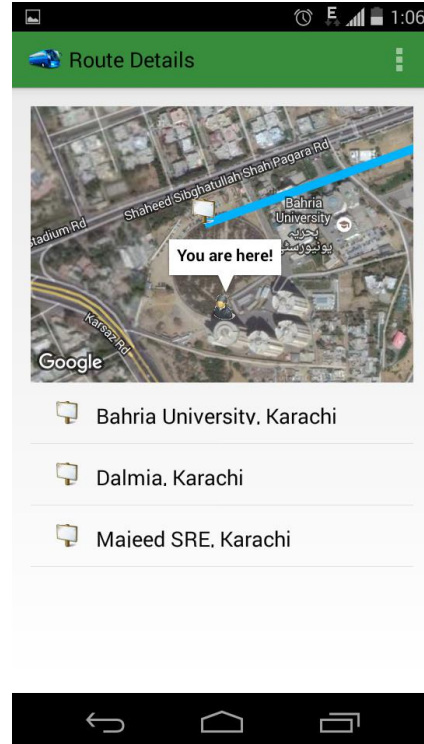


Figure 19 - Detail Activity View

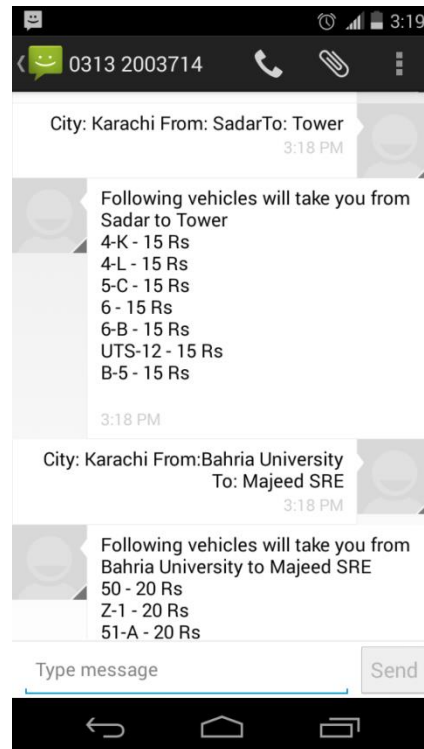


Figure 20 - Sample Response from SMS Service

Transit

Add Delete Settings

Add Vehicle Route Data Routes Vehicles Locations Settings

Vehicle Type	Chingche	Vehicle Name	Z-1
Route Fair	20	Stop Fair	15
Add Fairs			
Start Point	Millennium Mall, Karachi		
Latitude	24.861462	Longitude	67.009939
Add Start Point			
Next Bus Stop	Dhoraji Society, Karachi		
Latitude	24.887244	Longitude	67.076858
Add Bus Stop			
End Point	New Town Police Station, Karachi		
Latitude	24.928940	Longitude	67.114351
Add End Point			

```

</BusStop>
<BusStop>
  <BusStopName>Liaquat Hospital, Karachi
</BusStopName>
  <BusStopLat>24.892509</BusStopLat>
  <BusStopLong>67.068893</BusStopLong>
</BusStop>
<BusStop>
  <BusStopName>Dhoraji Society, Karachi
</BusStopName>
  <BusStopLat>24.887244</BusStopLat>
  <BusStopLong>67.076858</BusStopLong>
</BusStop>
<EndPoint>
  <EndPointName>New Town Police Station, Karachi
</EndPointName>
  <EndPointLat>24.928940</EndPointLat>
  <EndPointLong>67.114351</EndPointLong>
</EndPoint>
</Vehicle>
  
```

Format XML Generate Vehicle XML Generate Routes XML

Figure 21 - Transit Tool - Main XML View

Transit					
Add Delete Settings					
Add Vehicle Route Data Routes Vehicles Locations Settings					
All	Start Point	End Point	Route Data	Status	Upload
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Majeed SRE, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Dalmia, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Bahria University, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	National Stadium, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Agha Khan Hospital, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Liaquat Hospital, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	Dhoraji Society, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Millennium Mall, Karachi	New Town Police Station, Kara...	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	Dalmia, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	Bahria University, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	National Stadium, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	Agha Khan Hospital, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	Liaquat Hospital, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	Dhoraji Society, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Majeed SRE, Karachi	New Town Police Station, Kara...	<?xml version="1.0" encoding="UTF-8"?><R...		
<input checked="" type="checkbox"/>	Dalmia, Karachi	Bahria University, Karachi	<?xml version="1.0" encoding="UTF-8"?><R...		
Route Combinations 72 Upload Files Start Creating Routes Combination Files Files Created 0 Not Created 0					

Figure 22 - Routes Generation Window

TransitSMS					
Export Test Message About					Connected
Sender	Query	Start Point	End Point	Response	Replied
+923320307817	<message>...	Bahria Unive...	Majeed	Sorry we can't understand your destin...	True
+923320307817	<message>...	Sadar, Karachi	Tower, Kara...	Following vehicles will take you from ...	True
+923320307817	<message>...	Bahria Unive...	Majeed SRE...	Following vehicles will take you from ...	True

Figure 23 - Transit SMS Window

6.2. Conclusions

Transport is the backbone behind the largest GDP generating city Karachi, so the problems faced by 2 million of its commuters need to be addressed. Transit will play an important role in minimizing the problems related to information of routes & rides. Not only will this enhance the people's reliability on public transport but it will also increase the reach of public transport. After the success of Transit's system in Karachi, other cities around Pakistan will be incorporated into the system one by one.

6.3. Future work

We are planning to include following features to

1. Transit Pakistan in near future.
2. Including other cities of Pakistan
3. Real Time Monitoring
4. Estimated Time of Arrival
5. Bus Schedules
6. Including Train & Metro (where applicable)
7. iOS Applications.

Appendix A – Code, References

7.1. Code

Following is the code behind our Admin Dashboard of Transit Tool.

```

Using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;

namespace TransitTool
{
    public partial class MainForm : Form
    {
        Vehicles vehicles; //Object of Vehicle Class
        Locations locations; //Object of Locations Class
        XMLHandler xmlHandler; //object of XMLHandler class
        Filer filer; //Object of Filer class
        Settings settings; //Object of Settings class
        ArrayList globaldetails;

        public MainForm()
        {
            InitializeComponent();

            _InitilizeVariables();
        }

        private void _InitilizeVariables()
        {
            vehicles = new Vehicles();
            locations = new Locations();
            xmlHandler = new XMLHandler();
            filer = new Filer();
            settings = new Settings();
            globaldetails = new ArrayList();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {
            //method for Loading data in comoboboxes
            LoadComboBoxes();
        }

        //Load data in Comoboxes
        public void LoadComboBoxes()
        {
            //First clear all items in combo boxes if there are any.
            VehicleTypeComboBox.Items.Clear();
            StartPointNameComboBox.Items.Clear();
            EndPointNameComboBox.Items.Clear();
            DeleteVehicleTypeCBox.Items.Clear();
            AddNewVehicleVehicleTypeCBox.Items.Clear();
            DeleteVehicleVehicleTypeCBox.Items.Clear();
            UpdateLocNameCBox.Items.Clear();
            DeletelocNameCBox.Items.Clear();

            //Add Data in Vehicle Type Comobox

```

```

        VehicleTypeComboBox.Items.AddRange(vehicles.GetVehicleTypes());

        //Creating an array of Location Name.
        //Not doing it directly in comboboxes because that would take extra
iterations
        string[] LocationsArray = locations.GetLocationNames();

        //StartPoints Combobox Populating
        StartPointNameComboBox.Items.AddRange(LocationsArray);

        //BusStops Combobox Populating
        BusStopComboBox.Items.AddRange(LocationsArray);

        //EndPoints Combobox Populating
        EndPointNameComboBox.Items.AddRange(LocationsArray);

        DeleteVehicleTypeCBox.Items.AddRange(vehicles.GetVehicleTypes());

AddNewVehicleVehicleTypeCBox.Items.AddRange(vehicles.GetVehicleTypes());

DeleteVehicleVehicleTypeCBox.Items.AddRange(vehicles.GetVehicleTypes());

        UpdateLocNameCBox.Items.AddRange(locations.GetLocationNames());
        DeleteLocNameCBox.Items.AddRange(locations.GetLocationNames());

        UpdateFileJoinBox.Text = settings.GetFileJoin();
        UpdateFileExtensionBox.Text = settings.GetFileExtension();
        UpdateRootFolderBox.Text = settings.GetRootFolder();
    }

    //This method will be used to send back statuses according to its severity
    public void Status(string Message, int Severity)
    {
        //Severity should be from between 1-5
        //1-Success 2-Warning 3-Error 4-Remember 5-Important
        //tailor the message box according to severity

        MessageBox.Show(Message);
    }

    //To scroll down the routes richbox when new text added.
    Public void ScrollDown()
    {
        RoutesRichBox.ScrollToCaret();
    }

    private void addNewVehicleToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        Tabs.SelectedTab = VehiclesTab;
    }

    private void addNewVehicleTypeToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        Tabs.SelectedTab = VehiclesTab;
    }

```



```

        private void addNewLocationToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Tabs.SelectedTab = LocationsTab;
        }

        private void deleteVehicleToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            Tabs.SelectedTab = VehiclesTab;
        }

        private void deleteVehicleTypeToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Tabs.SelectedTab = VehiclesTab;
        }

        private void deleteLocationToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Tabs.SelectedTab = LocationsTab;
        }

        private void setFileJoinToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            Tabs.SelectedTab = SettingsTab;
        }

        private void setFileExtensionToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Tabs.SelectedTab = SettingsTab;
        }

        private void setRootFolderToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            Tabs.SelectedTab = SettingsTab;
        }

        private void VehicleTypeComboBox_SelectedIndexChanged(object sender,
EventArgs e)
        {
            //Clear if already populated data from Vehicle Name comobobox
            VehicleNameComboBox.Items.Clear();

            //Populate Vehicle Name comobobox with new selected vehicle type

            VehicleNameComboBox.Items.AddRange(vehicles.GetVehiclesByTypeName(VehicleTypeCombo
Box.Text));

            try
            {
                VehicleNameComboBox.Text = string.Empty;
                //Now Select a new first value by default I the previous vehicle
                type name is selected in combobox
            }
        }

```

```

        VehicleNameComboBox.SelectedIndex = 0;
    }
    catch
    {
        //This means there are no vehicles added to this vehicle type in
Database
        Status("No vehicle data available for " +
VehicleTypeComboBox.SelectedItem.ToString() + ".", 3);
    }

    RoutesRichBox.Text = xmlHandler.SetVehicleType(RoutesRichBox.Text,
VehicleTypeComboBox.SelectedItem.ToString());
    ScrollDown();
}

private void VehicleNameComboBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    RoutesRichBox.Text = xmlHandler.SetVehicleName(RoutesRichBox.Text,
VehicleNameComboBox.SelectedItem.ToString());
    ScrollDown();
}

private void AddFairsButton_Click(object sender, EventArgs e)
{
    RoutesRichBox.Text = xmlHandler.SetRouteFair(RoutesRichBox.Text,
RouteFairBox.Text, StopFairBox.Text);

    ScrollDown();
}

private void StartPointNameComboBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    decimal StartPointLatitude, StartPointLongitude;

    locations.GetCoordinates(StartPointNameComboBox.SelectedItem.ToString(), out
StartPointLatitude, out StartPointLongitude);

    StartPointLatBox.Text = StartPointLatitude.ToString();
    StartPointLongBox.Text = StartPointLongitude.ToString();
}

private void AddStartPointBtn_Click(object sender, EventArgs e)
{
    RoutesRichBox.Text = xmlHandler.SetStartPoint(RoutesRichBox.Text,
StartPointNameComboBox.SelectedItem.ToString(), StartPointLatBox.Text,
StartPointLongBox.Text);
}

private void BusStopComboBox_SelectedIndexChanged(object sender, EventArgs
e)
{
    decimal BusStopLatitude, BusStopLongitude;

    locations.GetCoordinates(BusStopComboBox.SelectedItem.ToString(), out
BusStopLatitude, out BusStopLongitude);

```

```

        BusStopLatitudeBox.Text = BusStopLatitude.ToString();
        BusStopLongitudeBox.Text = BusStopLongitude.ToString();
    }

    private void AddBusStopBtn_Click(object sender, EventArgs e)
    {
        //this variable is taken separately because we need to scroll down
        //the route rich box. Which I work without AppenText method.
        //So I took this 59eparately, then empty the rich box & send this
        value to our bus stop method.
        String RichTextBoxText = RoutesRichBox.Text;
        RoutesRichBox.Text = null;
        RoutesRichBox.AppendText(xmlHandler.SetBusStop(RichTextBoxText,
        BusStopComboBox.SelectedItem.ToString(), BusStopLatitudeBox.Text,
        BusStopLongitudeBox.Text));
        ScrollDown();
    }

    private void EndPointNameComboBox_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        decimal EndPointLatitude, EndPointLongitude;

        locations.GetCoordinates(EndPointNameComboBox.SelectedItem.ToString(),
        out EndPointLatitude, out EndPointLongitude);

        EndPointLatBox.Text = EndPointLatitude.ToString();
        EndPointLongBox.Text = EndPointLongitude.ToString();
    }

    private void AddEndPointBtn_Click(object sender, EventArgs e)
    {
        //this variable is taken separately because we need to scroll down
        //the route rich box. Which I work without AppendText method.
        //So I took this 59eparately, then empty the rich box & send this
        value to our bus stop method.
        String RichTextBoxText = RoutesRichBox.Text;
        RoutesRichBox.Text = null;
        RoutesRichBox.AppendText(xmlHandler.SetEndPoint(RichTextBoxText,
        EndPointNameComboBox.SelectedItem.ToString(), EndPointLatBox.Text,
        EndPointLongBox.Text));
        ScrollDown();
    }

    private void FormatXMLBtn_Click(object sender, EventArgs e)
    {
        try
        {
            RoutesRichBox.Text =
xmlHandler.IdentateXML(xmlHandler.FormatVehicleXML(RoutesRichBox.Text));
        }
        catch (Exception ee)
        {
            Status(ee.Message, 3);
        }
    }
}

```

```

private void GenerateVehicleXMLBtn_Click(object sender, EventArgs e)
{
    try
    {
        if (vehicles.HandleVehicleRouteXML(RoutesRichBox.Text,
xmlHandler.GetStartPointName(RoutesRichBox.Text),
xmlHandler.GetEndPointName(RoutesRichBox.Text),
xmlHandler.GetVehicleName(RoutesRichBox.Text),
xmlHandler.GetVehicleType(RoutesRichBox.Text)))
        {
            if (filer.CreateVehicleFile(RoutesRichBox.Text,
xmlHandler.GetVehicleName(RoutesRichBox.Text), settings.GetFileExtension()))
            {
                Status("Vehicle Data Added to Database", 1);
            }
            else
            {
                Status("Sorry! Unable to Add Vehicle Route to Database.",
1);
            }
        }
        else
        {
            Status("Sorry! Unable to Add Vehicle Route to Database.", 1);
        }
    }
    catch (Exception ee)
    {
        Status(ee.Message, 3);
    }
}

private void GenerateRoutesXMLBtn_Click(object sender, EventArgs e)
{
    FilesGridView.Rows.Clear();

    Tabs.SelectedTab = RoutesTab;

    string VehicleType = xmlHandler.GetVehicleType(RoutesRichBox.Text);
    string VehicleName = xmlHandler.GetVehicleName(RoutesRichBox.Text);
    string StopFair = xmlHandler.GetStopFair(RoutesRichBox.Text);

    RichBoxLines = RoutesRichBox.Lines;

    RouteGenerator.RunWorkerAsync(new Details() { VehicleType =
VehicleType, VehicleName = VehicleName, StopFair = StopFair });
}

string[] RichBoxLines;

private void RouteGenerator_DoWork(object sender,
System.ComponentModel.DoWorkEventArgs e)
{
    Details details = (Details)e.Argument;

    string VehicleType = details.VehicleType;
    string VehicleName = details.VehicleName;
    string StopFair = details.StopFair;

```

```

        e.Result = xmlHandler.GenerateRoutes(RichBoxLines, VehicleType,
VehicleName, StopFair);
    }

    private void RouteGenerator_RunWorkerCompleted(object sender,
System.ComponentModel.RunWorkerCompletedEventArgs e)
    {
        if (e.Error != null)
        {
            // handle the error

            Status(e.Error.Message + e.Error.StackTrace, 1);
        }
        else if (e.Cancelled)
        {
            // handle cancellation
        }
        else
        {
            ArrayList DataList = (ArrayList)e.Result;

            RouteCombinationsLabel.Text = DataList.Count.ToString();

            foreach (Details details in DataList)
            {
                FilesGridView.Rows.Add(true, details.StartPoint,
details.EndPoint, details.Content);
            }
        }
    }

    private void FilesGridView_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {
        if (e.RowIndex >= 0)
        {
            if (FilesGridView.Columns[e.ColumnIndex].Name ==
"RouteDataColumn")
            {
                Status(FilesGridView.Rows[e.RowIndex].Cells["RouteDataColumn"].Value.ToString(),
1);
            }
            else if (FilesGridView.Columns[e.ColumnIndex].Name ==
"StatusColumn")
            {
                BackgroundWorker RouteFileCreator = new BackgroundWorker();
                RouteFileCreator.DoWork += new
DoWorkEventHandler(RouteFileCreator_DoWork);
                RouteFileCreator.RunWorkerCompleted +=
(RouteFileCreator_RunWorkerCompleted);
                RouteFileCreator.RunWorkerAsync(new Details() { RowNumber =
e.RowIndex, VehicleName =
xmlHandler.GetVehicleName(FilesGridView.Rows[e.RowIndex].Cells["RouteDataColumn"].
Value.ToString()), Content =
FilesGridView.Rows[e.RowIndex].Cells["RouteDataColumn"].Value.ToString(),
StartPoint =

```

```

FilesGridView.Rows[e.RowIndex].Cells["StartPointColumn"].Value.ToString(),
EndPoint =
FilesGridView.Rows[e.RowIndex].Cells["EndPointColumn"].Value.ToString(),
RootFolder = settings.GetRootFolder(), Join = settings.GetFileJoin(), Extension =
settings.GetFileExtension() });
    }
}

private void RouteFileCreator_DoWork(object sender,
System.ComponentModel.DoWorkEventArgs e)
{
    Details details = (Details)e.Argument;

    e.Result = (new Details() { FileCreatedStatus =
filer.CreateRouteFile(details.VehicleName, details.Content, details.StartPoint,
details.EndPoint, details.RootFolder, details.Join, details.Extension), RowNumber
= details.RowNumber });
}
private void RouteFileCreator_RunWorkerCompleted(object sender,
System.ComponentModel.RunWorkerCompletedEventArgs e)
{
    if (e.Error != null)
    {
        // handle the error

        Status(e.Error.Message + e.Error.StackTrace, 1);
    }
    else if (e.Cancelled)
    {
        // handle cancellation
    }
    else
    {
        Details details = (Details)e.Result;

        FilesGridView.Rows[details.RowNumber].Cells["StatusColumn"].Value
= details.FileCreatedStatus;

        if (details.FileCreatedStatus == "Not Created")
        {
FilesGridView.Rows[details.RowNumber].DefaultCellStyle.BackColor = Color.Red;
        }
        else if ((details.FileCreatedStatus == "Created") ||
(details.FileCreatedStatus == "Merged"))
        {
            if
(FilesGridView.Rows[details.RowNumber].DefaultCellStyle.BackColor == Color.Red)
            {

FilesGridView.Rows[details.RowNumber].DefaultCellStyle.BackColor =
Color.FromArgb(76, 175, 80);
            }
        }
    }
}
}

```

```

private void StartCreatingFilesButton_Click(object sender, EventArgs e)
{
    RoutesIterator.RunWorkerAsync();
}

private void RoutesIterator_DoWork(object sender, DoWorkEventArgs e)
{
    foreach (DataGridViewRow Row in FilesGridView.Rows)
    {
        if
(Convert.ToBoolean(FilesGridView.Rows[Row.Index].Cells["SelectAllColumn"].Value))
        {
            BackgroundWorker RouteFileCreator = new BackgroundWorker();
            RouteFileCreator.DoWork += new
DoWorkEventHandler(RouteFileCreator_DoWork);
            RouteFileCreator.RunWorkerCompleted +=
(RouteFileCreator_RunWorkerCompleted);
            RouteFileCreator.RunWorkerAsync(new Details() { RowNumber =
Row.Index, VehicleName =
xmlHandler.GetVehicleName(Row.Cells["RouteDataColumn"].Value.ToString()), Content
= Row.Cells["RouteDataColumn"].Value.ToString(), StartPoint =
Row.Cells["StartPointColumn"].Value.ToString(), EndPoint =
Row.Cells["EndPointColumn"].Value.ToString(), RootFolder =
settings.GetRootFolder(), Join = settings.GetFilesJoin(), Extension =
settings.GetFilesExtension() });
        }
    }
}

private void RoutesIterator_RunWorkerCompleted(object sender,
RunWorkerCompletedEventArgs e)
{
    if (e.Error != null)
    {
        // handle the error

        Status(e.Error.Message + e.Error.StackTrace, 1);
    }
    else if (e.Cancelled)
    {
        // handle cancellation
    }
}

private void AddSelectedVehicleTypeBtn_Click(object sender, EventArgs e)
{
    if (vehicles.AddNewVehicleType(AddVehicleTypeBox.Text))
    {
        Status("Vehicle Type Added to Database.", 1);

        _InitilizeVariables();
        LoadComboBoxes();
    }
}

private void DeleteSelectedVehicleTypeBtn_Click(object sender, EventArgs
e)
{

```

```

        if
(vehicles.DeleteVehicleType(DeleteVehicleTypeCBox.SelectedItem.ToString()))
        {
            Status("Vehicle Type Deleted from Database.", 1);

            _InitilizeVariables();
            LoadComboBoxes();
        }
    }

    private void AddNewVehicleBtn_Click(object sender, EventArgs e)
    {
        if (vehicles.AddNewVehicle(AddNewVehicleNameBox.Text,
vehicles.GetVehicleTypeID(AddNewVehicleVehicleTypeCBox.SelectedItem.ToString())))
        {
            Status("New Vehicle added to Database.", 1);

            _InitilizeVariables();
            LoadComboBoxes();
        }
    }

    private void DeleteVehicleVehicleTypeCBox_SelectedIndexChanged(object
sender, EventArgs e)
    {
        DeleteVehicleNameCBox.Items.AddRange(vehicles.GetVehiclesByTypeName(DeleteVehicleV
ehicleTypeCBox.SelectedItem.ToString()));
    }

    private void DeleteSelectedVehicleBtn_Click(object sender, EventArgs e)
    {
        DeleteVehicle DeleteVehicleForm = new
DeleteVehicle(DeleteVehicleNameCBox.SelectedItem.ToString(),
DeleteVehicleVehicleTypeCBox.SelectedItem.ToString());

        DeleteVehicleForm.ShowDialog();
    }

    private void AddNewLocBtn_Click(object sender, EventArgs e)
    {
        if (locations.AddNewLocation(AddLocNameBox.Text,
Convert.ToDecimal(AddLocLatitudeBox.Text),
Convert.ToDecimal(AddLocLongitudeBox.Text)))
        {
            Status("New Location Details added to Database.", 1);

            _InitilizeVariables();
            LoadComboBoxes();
        }
    }

    private void UpdateLocNameCBox_SelectedIndexChanged(object sender,
EventArgs e)
    {
        decimal LocationLatitude, LocationLongitude;

```



```

        locations.GetCoordinates(UpdateLocNameCBox.SelectedItem.ToString(),
out LocationLatitude, out LocationLongitude);

        UpdateLocLatitudeBox.Text = LocationLatitude.ToString();
        UpdateLocLongitudeBox.Text = LocationLongitude.ToString();
    }

    private void UpdateSelectedLocBtn_Click(object sender, EventArgs e)
    {
        int WhoShouldBeUpdated = 0;

        if (UpdateLocationNameRBtn.Checked)
        {
            WhoShouldBeUpdated = 1;
        }
        else if (UpdateLocationLatitudeRBtn.Checked)
        {
            WhoShouldBeUpdated = 2;
        }
        else if (UpdateLocationLongitudeRBtn.Checked)
        {
            WhoShouldBeUpdated = 3;
        }

        if (locations.UpdateLocation(UpdateLocNameCBox.Text,
Convert.ToDecimal(UpdateLocLatitudeBox.Text),
Convert.ToDecimal(UpdateLocLongitudeBox.Text), WhoShouldBeUpdated))
        {
            Status("Selected Location Details Updated.", 1);

            _InitilizeVariables();
            LoadComboBoxes();
        }
    }

    private void DeleteLocNameCBox_SelectedIndexChanged(object sender,
EventArgs e)
    {
        decimal LocationLatitude, LocationLongitude;

        locations.GetCoordinates(DeleteLocNameCBox.SelectedItem.ToString(),
out LocationLatitude, out LocationLongitude);

        DeleteLocLatitudeBox.Text = LocationLatitude.ToString();
        DeleteLocLongitudeBox.Text = LocationLongitude.ToString();
    }

    private void DeleteSelectedLocBtn_Click(object sender, EventArgs e)
    {
        decimal LocationLatitude, LocationLongitude;

        locations.GetCoordinates(DeleteLocNameCBox.SelectedItem.ToString(),
out LocationLatitude, out LocationLongitude);

        if
(locations.DeleteLocation(DeleteLocNameCBox.SelectedItem.ToString(),
LocationLatitude, LocationLongitude))
        {

```

```
        Status("Selected Location Details Deleted.", 1);

        _InitilizeVariables();
        LoadComboBoxes();
    }
}

private void UpdateFileJoinBtn_Click(object sender, EventArgs e)
{
    if (settings.SetFileJoin(UpdateFileJoinBox.Text))
    {
        Status("File Join Setting Updated.", 1);

        _InitilizeVariables();
        LoadComboBoxes();
    }
}

private void UpdateFileExtensionBtn_Click(object sender, EventArgs e)
{
    if (settings.SetFileExtension(UpdateFileExtensionBox.Text))
    {
        Status("File Extension Setting Updated.", 1);

        _InitilizeVariables();
        LoadComboBoxes();
    }
}

private void SelectRootFolderDialogBtn_Click(object sender, EventArgs e)
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();

    if (fbd.ShowDialog() == DialogResult.OK)
    {
        UpdateRootFolderBox.Text = fbd.SelectedPath;
    }
}

private void UpdateRootFolderBtn_Click(object sender, EventArgs e)
{
    if (settings.SetRootFolder(UpdateRootFolderBox.Text))
    {
        Status("Root Folder Setting Updated.", 1);

        _InitilizeVariables();
        LoadComboBoxes();
    }
}
}
```

7.2. References

Dawn News Publication. <http://www.dawn.com/1158772>. 23 Feb 2015.

Dawn News Publication. <http://www.dawn.com/news/1158772>. 23 January 2015.

Du, Zhenjun. "Mobile Location Service Design Based on Android." *5th International Conference on Wireless Communications, Networking and Mobile Computing* (2009): 1-4.

Qadeer, Mohammed Abdul. "The personal SMS gateway." *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference* (2011): 617 - 621.

Quinstreet Enterpris. <http://www.webopedia.com/TERM/A/API.html>. 22 March 2015.