Ceylon Bus Routes: A Bus Route Guide

Author name with initials
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.
Type your email here

***Abstract*— Mobility is a basic human requirement for traversing between different locations. Public transport bus service fulfills this requirement by operating according to a published timetable along a route calling at agreed bus stops. Despite its progress at a rapid pace, moving commuters are riddled with problems of finding the correct bus route to reach their desired destinations. The use of mobile phone applications is an attractive alternative to guide passengers through correct bus routes. This paper discusses the project, "Ceylon Bus Routes" which is a Microsoft Windows universal application built as a solution for finding bus routes in Sri Lanka. This application is significant because it works offline and it can be run on any Windows based device. The goal of the application is to display the best routes as well as all the routes between two locations requested by the user. The application also uses GPS technology to track user location in case if the user does not know his or her actual location.**

Keywords—Windows; Universal application; Bus Route; XAML; GPS; Start to Destination; Model View ViewModel

I. INTRODUCTION

Public bus transport service still holds the primary mode of transportation due to burgeoning population and traffic problems in most countries. This has ended up with the problem of finding the correct bus route to reach the destination for many travelers. Technology can provide a great support towards the use of public transport, by providing sufficiently accurate information in a user friendly manner. Suppose that a certain traveler wants to visit some important cities of the country, yet a map would not provide sufficient information for that. Asking from a person does not assure the journey completely. Hence 'Ceylon Bus Routes' application offers a very accurate and easy solution to this problem by providing information about the sequence of buses to reach the required destinations.

Recently, much of the growing interest has been pursued in the context of building Windows universal applications instead of sole mobile applications for Windows phone. Although the Windows Phone Store is catching up, it is still very much behind Android and iOS App Stores which stranglehold on the smartphone market exceeds 90% [1]. One of the key drivers behind a mobile or desktop platform's success is the number of applications available in its app store. From this perspective, Windows App Store does not contain enough applications. Microsoft has acknowledged the difficulty of developing applications separately for Windows Phone Store and Windows Store. Therefore they came up with the solution of universal apps.

Ceylon Bus Routes is an offline application that facilitates commuters to find bus routes in Sri Lanka. It is built in the form of a universal app. This application can be run in devices, which has an operating system newer than Windows 8. This is initially designed for Sri Lanka, but it can be enhanced to cover any part of the world. Hence the goal of the application is to provide a useful tool to moving commuters, by calculating best routes and guiding them through a mobile application.

This paper describes how Ceylon Bus Routes application was designed and developed. Section II surveys related work in building applications for finding bus routes. Section III provides an overview of the system. System models, design, implementation and testing of the application are described in latter sections respectively. Final section of the paper describes conclusion and future work.

II. LITERATURE REVIEW

This section provides an overview of some applications that are available in different platforms to find bus routes.

The most rated Android application to find bus routes in Sri Lanka is "Colombo Bus Route" offered by Pirinthapan. With this, users are able to find most of the available bus routes within Colombo district limits [2].

Another Android application "Colombo Bus Route" offered by TechWire also provides a similar service to reach towns within Colombo area in Sri Lanka. The developers are planning in upcoming updates to sort the options according to shortest distance first in case where multiple bus routes are available to be taken, otherwise the user has to go through each to find the best method [3].

"Sri Lanka Bus Route" offered by Hseya is another Android application which provides details about bus routes in Sri Lanka. According to the ratings and user feedback, the execution of this application is unsuccessful. Some people claim that it is a half cooked app because it does not contain enough information about the bus routes. This application requires a better standard of usability too [4].

"Busfinder" is another interesting work that provides a personalized multimodal transportation mobile guide, which was initially designed for the city of Athens and Greece. It utilizes dynamic routing algorithms, knowledge engineering principles and Semantic Web technologies [5].

Another smart location-aware application for bus guide in Seoul based on GPS recognizes the current position with GPS

receiver and guides users to the nearby bus stops. It provides the users with bus information at the bus stops [6].

A user centric geospatial decision support tool has been developed for bus travel planning based on design and modeling considerations that are more individual centric. The distinctive characteristic of this tool is the incorporation of passengers' cognition information processing and decision making process during their search for a bus travel plan [7].

Many transportation guides have been implemented so far. However, Windows Store does not have a proper application to find bus routes yet. Ceylon Bus Routes is a solution that is built to overcome this problem. It differs in several aspects from the aforementioned applications. Firstly, an offline app would be very useful because internet connections are not available all the time at all locations in a country. On the other hand, online applications are expensive and time consuming too. GPS is an additional facility for the user to find his or her actual location and to find the route accordingly. The most significance of this application lies in the fact that this covers all of the bus routes throughout the country. Current applications have several limitations. No application has still been developed in any mobile platform to cover all the bus routes in Sri Lanka. They are unable to deliver search results if the route between the start location and the destination is too long. In fact, they are unable to deliver route details for journeys which require more than two buses, whereas Ceylon Bus Routes is capable to deliver search results up to any number of buses by defining it in the algorithm.

III. Overview Of The System And Process Followed

Rational Unified Process was used as the process model, which was based on four main phases; inception (where most of the planning work was done), elaboration (where most of the design work was done), construction (where the real implementation was taken place) and transition (where deployment and maintenance work were carried out). The elaboration phase and the construction phase were focused on three main iterations, which included creating the data file covering all bus routes in Sri Lanka, calculating route from start location to destination and tracking current location using GPS. During the inception phase, the business model and the requirements of the project were specified. System architecture was designed in the elaboration phase. Construction phase was used for implementation of the features in accordance to the system architecture with proper testing. During the transition phase, refactoring of the code, system testing and deployment were mainly carried out.

IV. System Models

A. System Requiremnt Specification

This section describes the system requirement specification, which was launched at the inception phase of the project.

'Start to Destination', 'Current location to Destination', 'Buses at a Location' and 'All Routes' are identified as the four main functional requirements. The ultimate goal of all these functionalities is to guide the passengers by providing appropriate instructions for their queries. The application has considered one directional and bi directional routes separately to deliver most accurate solutions for moving commuters. Search results are displayed as 'Best routes' as well as 'All routes' to provide a better user experience. 'Buses at a Location' displays all the bus numbers that passes a particular location. 'All Routes' page is a reference that displays the sequence of bus stops in different routes.

An application that provides services to public commuters faces several challenges. User-friendliness, accuracy, reliability, performance, simplicity and provision of comprehensive instructions are some of them. Usability has been optimized with in this application, so that any smart phone user can easily operate this application. A high degree of usability is achieved through uncluttered interfaces with a pleasant look and feel. The interfaces follow the same layout and patterns as other Windows applications. The consistency across different pages is achieved by using the same layouts. The icons used are compatible with other Windows applications to ensure that the user will be able to understand easily what they mean. Auto focusing is included to make the searching process fast and easy. Auto suggestion boxes are added for user inputs, so that the user might not enter locations that are not available in the database. The interfaces guide the users to achieve the desired service. Whenever a user performs an invalid action, message boxes are popped up to explain the error. 'Help' option can be referred when the user is not clear about following correct procedures to get the service. Therefore the system is easy to learn and handle.

Search results should be accurate so that the passengers are not misled. Reliability guarantees that an accurate service is provided within a less time at a less cost. There are no possibilities of any failure unless the device gets switched off.

The application should avoid the user waiting a long time to retrieve the results of a search. The task time taken to provide search results should be less than half a minute. Latencies can be expected at times when tracking current location using GPS. The response of UI elements to click events should not exceed half a second. Asynchronous programming is used to achieve a high standard of performance by ensuring that the application remains responsive when it does other work that might take an extended amount of time. Hence loading and processing the data file is done via asynchronous programing in which a function call is issued by the application when the task is to start and it will return to the application once the task is completed.

B. Use Case Diagram

Mainly, a user can enter the start location and a destination the the system displays all possible routes between two locations. The all routes information includes the number of buses up to a maximum of four that can be taken for each route. Best routes with minimum number of buses are displayed through 'Best routes' option. Also the system can get the user's current location using GPS technology and display the routes for a given destination. Another function is displaying all the bus (route) numbers that pass a particular location given by a user. Also, the system provides a reference

to find out the bus stops in different routes. Each line shows the details of a certain bus route with the number of the bus route followed by the sequence of bus stops that belongs to that route. In-route and out- route are shown separately in case of one directional routes. Further, help service provides the user with the basic guidelines to achieve different services of the application. Main functionalities of the application are shown in Fig. 1.

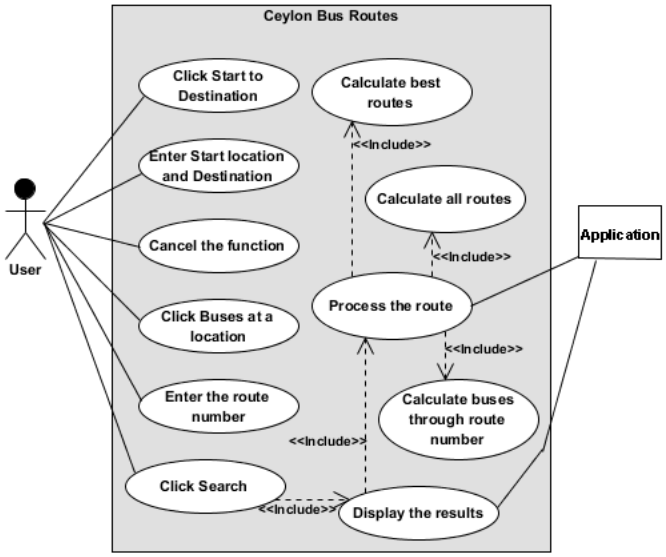

Fig. 1. Use case diagram of the system

V. System Design

A. Architecture of the system

The system follows a three-tier architecture as shown in Fig. 2. It consists of three layers; a view layer on top, a business logic layer in the middle, and a data access layer at the bottom. Each layer exposes an interface to the layers above it. This type of architecture increases the abstraction levels and provides a standardization of layer interfaces for libraries, frameworks and design patterns used. The view layer (presentation layer) consists of the interfaces with which the user interacts with the application. Business logic layer (domain Layer) contains the core of the application, where all processing and manipulation activities happen. An algorithm is implemented within this layer to calculate the route from one location to another location. Data access layer is responsible for retrieving data from the data file to calculate route. All these three layers communicate with each other to generate accurate search results eventually.

For the ease of implementing the above architecture within the system, Model View design pattern is used with XAML platform and its intent is to provide a clean separation between the user interface controls and their logic. The Model, View and ViewModel have distinct and separate roles. At higher level, View knows about the ViewModel and ViewModel knows about the Model. But Model is unaware of the ViewModel and the ViewModel is unaware of the View. Model represents the actual data or information we are dealing with. View is responsible for encapsulating the user interface and its logic. ViewModel handles the presentation logic and the state while acting as the mediator between the View and the Model.

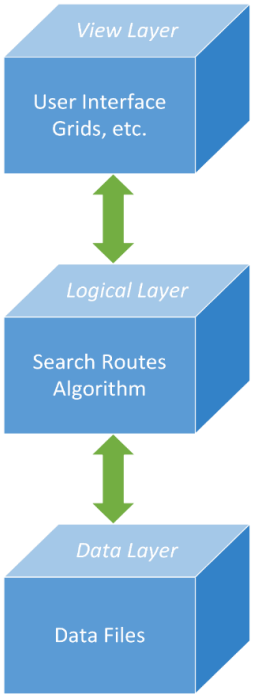

Fig 2. Layered architecture of the system

B. Logical View

The class diagram for Start to Destination functionality is shown in Fig. 3. Each class is an element of Model, View or ViewModel. There are separate classes in View component for each service that the application provides. These classes, which are inherited from Page class initialize the user interface components and handle click events.

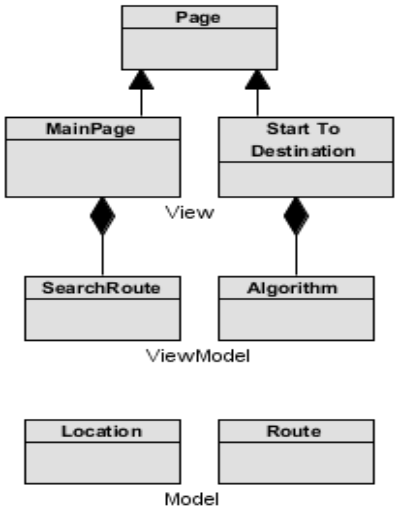

Fig. 3. Class diagram for Start to Destinatiion functionality

Model classes store data about the locations and routes extracted from the data file. Each location and route has an object associated with it. The Route object has a route number, two linked lists of Location objects for in-route and out-route because one directional and bi directional routes are required to be taken into consideration separately when implementing the algorithm.

ViewModel includes two classes 'SearchRoute' and 'Algorithm'. 'SearchRoute' class is responsible for loading the data file, reading it using asynchronous programming and inserting the data into two dictionaries containing Location and Route objects prior to display of the main page. This approach is useful to prevent reloading the data file whenever a new action or service on the main page is called. Use of dictionaries in this context helps to find objects by user defined keys. 'Algorithm' class implements the algorithm.

C. Process View

The sequence diagram for Start to Destination functionality is shown in Fig. 4. This specifies object interactions arranged in time sequence.

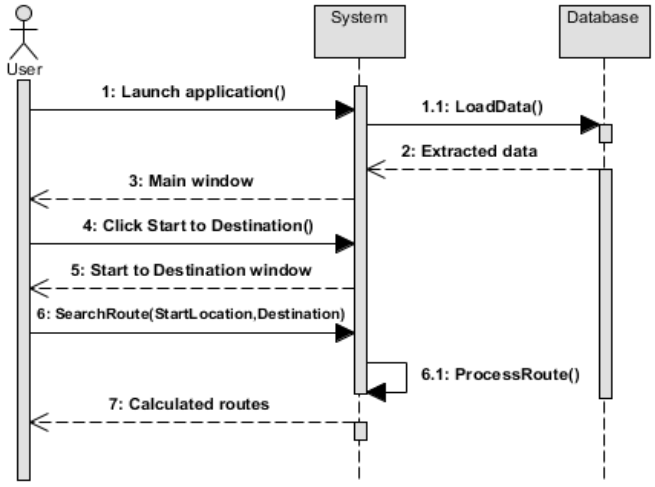

Fig. 4. Sequence diagram for Start to Destination functionality

When user launches the application, system loads the database and extracts all the route information available within the data file. The main window is returned after the data file is successfully processed. The user can direct to different types of services from the main page. When user requests the route information between two locations, the system calculates the route using a specific algorithm and returns the results.

D. Database design

The application stores data about the bus routes in a text file. Database schema of the system has a single entity which contains detials about each route. Route has three attributes called route number, in-route and out-route where primary key is the route number. Data is retrieved to the application by reading the data file via asynchronous programming. Current database of the application contains details about main bus routes all over the country.

VI. System Implementation

A. Implementation procedure

The requirement analysis, designing, coding, refctoring and testing were done in parallel through out the project.The implementation of the application was done using C# and XAML. The views of the user interfaces were coded using XAML and the implementation logic was coded using C#. Microsoft Visual Studio integrated with Windows SDK was used as the IDE for coding because it supports .NET framework and it has plenty of tools to build, deploy and test the system. Microsoft Visual Studio, which is the premier professional design tool for XAML applications, integrates features for development of universal applications. That allows making the application to work in different Windows devices with the same codebase for implementing the logic but different views at runtime. Coding, debugging, testing and diagnosing issues are fast and easy with Microsoft Visual Studio. Also, it consists of advanced syntax highlighting, IntelliSense code completion and intuitive source navigation in the code editor. Code base can be optimized with powerful refactoring and duplicate code detection tools. Tools like CPU usage, memory usage and app timeline in the performance and diagnostics hub were used to identify performance related bottlenecks of the application.

The Windows Software Development Kit (SDK) for Windows 8.1 contains headers, libraries and tools that can be used to create applications running on Windows operating systems. The Windows SDK includes the Windows App Certification Kit 3.3 to test apps for Windows 8.1 and Windows 8.1 Phone certification program to create applications for platform chipsets X86, X64 and Itanium with .NET Framework [8]. The project consists of three main sub projects specifically for the Windows phone application, Windows desktop application and another shared sub project for including the common codebase of both aforementioned projects. Having this type of shared project structure enables developing universal applications easily without any repetition of code. After coding each step, the project was first built and errors were diagnosed. Testing had been carried out using Simulator, Virtual machine, Emulators and real Windows phones. Code analysis had been carried out often to detect defects and errors. Quality of the code had been analyzed using code metrics analysis.

Designing the database was a very precise and important activity because accuracy of the search results lies on the accuracy of the algorithm as well as on the data in database. If a certain location is included in the data file at some route, it should be included in all routes (buses) passing that particular location. That makes a consistent database, which is an important requirement of the application. Sequences of bus stops in different routes have also been manually checked using maps to accomplish an accurate and consistent database.

B. Algorithm

The application calculates the route between two locations using a tree search algorithm. It is a recursive algorithm that can find routes up to 'n' number of buses. Results that contain more than 'n' number of buses are ignored to make it more practical. The pseudo code of the algorithm is shown in Fig. 5.

```
loc ← Start location
dest ← Destination
depth ← n
FindRoute(loc , depth){
   IF (depth > 0 ) {
       routesList ← list of routes across loc
       FOR EACH (route in routesList) {
            IF (route contains dest) {
               nextLocation ← move to next
                              location
                                    through
                              that route
               loc ← nextLocation
               FindRoute(loc , depth -1)
            }
            ELSE {
               move to next route in routesList
            }
      }
   }
```

Fig. 5. Pseudo code of the serach algorithm

The process of finding the destination continues through all routes that pass across all locations from start location to destination as well as through all routes across all the new locations found in those respective routes. This is to generate all possible solutions of routes from start location to destination. Journeys that require minimum number of routes are sorted out to generate best solutions.

C. Main Interfaces

The interfaces for the mobile application and desktop application were designed separately by integrating features of universal applications.

When the application is launched, the user is directed to the main page (mobile version) in Fig. 6. Main page displays the four main services. Clicking 'Start to Destination' option directs the user to the corresponding page as in Fig. 7. Start location and destination are designed as auto suggestion boxes that suggest all the locations in the data file. After entering the start location in the relevant text box, the interface auto focuses to the destination text box making the search process easy and fast. Search results can be found as 'best routes' as well as 'all routes'. All these techniques enhance the user experience of the product.

Current application displays solutions containing four routes by maximum. If a particular journey requires more than four buses, application will display the message 'Journey is too long'. 'Buses at a Location' page contains a text box that suggests all available locations in the database. User can set the location and find the bus numbers, which pass that location. 'All Routes' page displays information about all available bus routes within the application. Map focuses at the current location. Slider zooms in and out within that area.

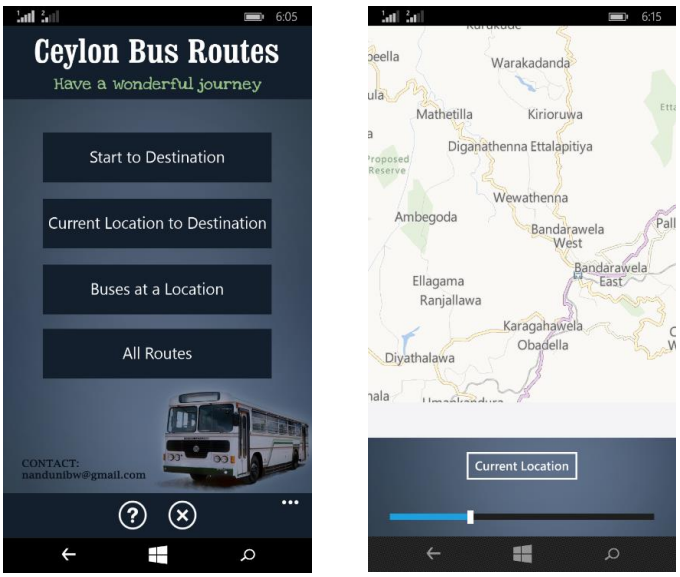

Fig. 6. Main page and Map of the mobile version

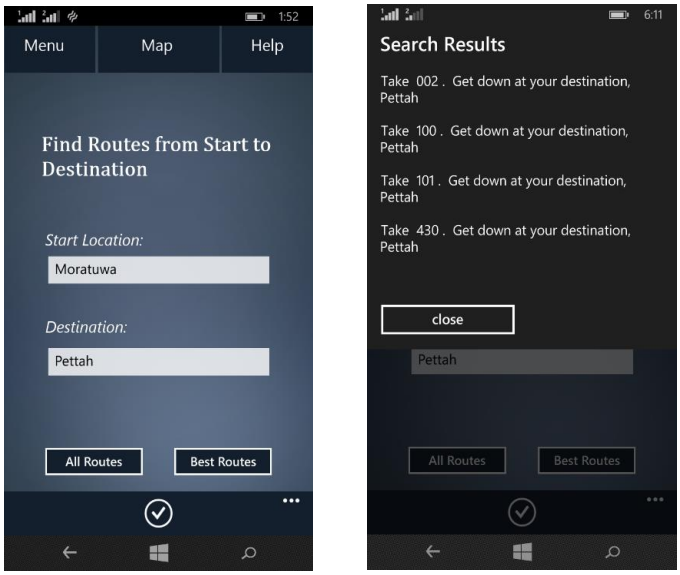

Fig. 7. Start to destination page and search results

VII. System Testing And Analysis

The mobile application was tested using Emulator 8.1 WVGA 512 MB, which is a desktop application that simulates a Windows Phone device through a virtualized environment to debug and test apps without a physical device. This provides similar performance to a physical Windows Phone device. The application was often tested with different types of Emulators as well as with real Windows phones. The desktop version of the application was tested using Windows simulator and the Windows virtual machine.

Unit testing is a testing process in which the smallest testable parts of an application are independently tested for proper operation. Automated and manual unit testing were carried out using Unit Test Generator and Microsoft Unit

Testing framework, respectively. Test Explorer is used to integrate unit tests and Fig. 8 shows the results obtained.

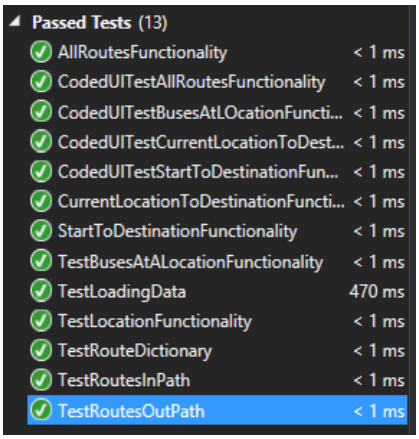

Fig. 8. Results of unit testing using Test Explorer

Coded UI testing is an extension used for testing the functionalities of user interface controls. Different graphical user interface controls were selected using a cross hair tool, added to a user interface control map and a series of actions were recorded and tested. Testing using Code Analysis Window was useful to manage and resolve code analysis warnings. Static code analysis was used to analyze source code prior to the execution in order to detect issues on codebase, identify potential design, interoperability, performance and security. This enhanced the quality of software by examining common defects and violations. Testing using Code Clone Analyzer was useful to search duplicate code throughout the solution. This analyzer detects direct copies and fragments, which differ in names of variables and parameters and in which some statements have been rearranged.

Beta testing was carried out by several volunteer Windows phone users. One of the major problems raised there was the screen compatibility due to different resolutions in different devices. Therefore, the orientation of widgets in the user interfaces were rearranged by modifying the related XAML files to eliminate this issue.

Code metrics analysis makes the code more maintainable by detecting where code needs rework. This included analyzing maintainability index, cyclomatic complexity, depth of inheritance and class coupling. Having the maintainability index between 20 and 100 indicates that the code has good maintainability, which means the code is easy to maintain. Low coupling and high cohesion are good since they indicate that the design is easy to reuse and maintain. The code was refactored often to achieve a better standard of quality.

Performance and Diagnostic Analyzer measures the user interface responsiveness, which makes the application fast and fluid. As shown in Fig. 9, this also includes analyzing memory usage, energy consumption and CPU usage. Long time consumption when loading the data file was detected using this analyzer. Hence, asynchronous programing was used as a remedy.

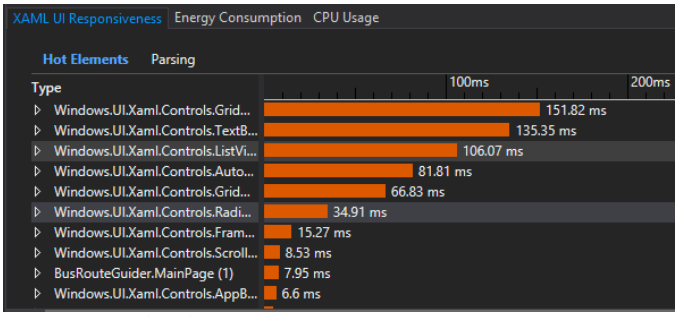

Fig. 9. Results of Performance and Diagnostic Analyzer

VIII. Conclusion And Future Work

The paper has described the design and implementation of a Windows based offline application to find bus routes in Sri Lanka. The application provides a successful solution to the unavailability of an application in Windows store to help moving commuters to find the bus routes. This application provides many functionalities including display all routes, finding best routes, details about different routes and bus stops. A broad coverage of routes all over the country enhances the productivity and accuracy of the application. The system is developed with the consideration of device compatibility, screen compatibility, user experience, stability and information accuracy.

This application is implemented using design patterns and software libraries enabling extensions in future upgrades. The application can be extended to cover all routes with all bus stops and major locations in Sri Lanka. Also, new functionalities such as displaying distance between bus stops and the fare. Further, visualization techniques can be improved to display the route from a start location to a destination with a side map along with the search results.

References

[1] Windows Phone's big problem, TNW News, [online], Available: http://thenextweb.com/microsoft/2013/09/14/windows-phone/ [Accessed: 12 September 2015].

[2] Colombo Bus Route, Google play, [online], Available: https://play.google.com/store/apps/details?id=cbr.arima&hl=en [Accessed: 13 September 2015].

[3] Colombo Bus Route – Android App, TechWire, [online], Available: http://techwire.lk/colombo-bus-route-android-app/ [Accessed: 13 September 2015].

[4] Sri Lanka bus route – Android, AndroSmartApps, [online], Available: http://www.androsmartapps.com/srilanka/Android-Sri-lanka-bus-route.html [Accessed: 13 September 2015].

[5] Tsolkas, Dimitris, et al. "Busfinder: a personalized multimodal transportation guide with dynamic routing." *Informatics (PCI), 2012 16th Panhellenic Conference on*. IEEE, 2012.

[6] Choi, Joo-Yen, et al. "A Smart Location-Aware Application for Bus Guide based on GPS." *International Journal of Information Processing and Management* 2.1 (2011): 101-108.

[7] Zhou, Yuhong, Jean-Claude Thill, and Zhengdong Huang. "Design of a user-centric decision support tool for fixed-route bus travel planning." *Applied Geography* 31.3 (2011): 1173-1184.

[8] Windows Software Development Kit (SDK) for Windows 8.1, Microsoft, [online], Available: https://msdn.microsoft.com/en-us/windows/desktop/bg162891.aspx [Accessed: 14 September 2015].