Software Requirement specification

Authors
Levi Goldfein 1257360
Sabeehah Ismail 797621
Sergio Oliveira 1101482
Storm Menges 1102107

August 28, 2017

Project : Shopping Route Recommender.

1 Product Scope

Our system is a shopping route recommender (SRRec) that maps out a route of the nearest shops in the user's area. The user will identify the items that they need to go shopping for, which could be groceries, clothing, appliances, etc. The items will be logged into a shopping list. On the user's required shopping day, they will run the SRRec and obtain a recommended route of shops to follow in a particular order to purchase the items they listed.

The SRRec will be a web-application and therefore should run on any supported web browser that has online access. The database stores each shop in the area with the items that they sell, and their respective prices.
Not only will the SRRec recommend a route of shops to purchase the user's required items, but it may also provide a route based on the shortest path for the minimal total expenses or the shortest travel time.

The benefits of the SRRec are the following:
-The user is given a recommended route of the nearest shops in the area to purchase their required items.
-If the user wants a route to purchase their items at the cheapest prices, the SRRec will provide a route based on the shortest path for the minimal total expenses.
-If the user wants a route that will allow them to purchase their items in the least amount of time, the SRRec will provide a route based on the shortest travel time.

1.1 Purpose

The purpose of the project is to create quality software that allows the user to enter items they require and the software will generate the shortest path to go and buy the items they

require. Our main goal is to produce quality software that is relevant and provides an accurate solution to all users at any given time

1.2 Product Overview

Our product is a software product in which we will allow the user to input items that they require and we will generate a route which will take the shortest amount of time, Our product is most likely to target working class people who dont have much time to go to the shops. Our product will also include various help buttons and prompts to guide users on how to use the software

1.3 Product Function

Our product will have one main function which is allowing the user to input items that they require and generating a route which will take the shortest amount of time, we would also like to add a lowest cost route which will take into account the petrol spend as well as the prices of the items bought

1.4 User Classes and Characteristics

Our user class will be a general user with focus on people who have heavy time constraints with regards to work

1.5 Operating Environment

The operating environment will be majority of browsers as we will integrate support for most big browsers, the coding environment will be Visual Studio and the Language will be C-Sharp

1.6 Assumptions and Dependencies

Our assumptions and dependencies include the fact that we will receive databases of items and their various prices from various supermarkets around Gauteng. We also assume that all price changes will be updated regularly by the supermarkets and we will receive those changes as they happen. To simulate this we will create our own scaled down databases which will contain a few products and their respective prices

2 Intended User and User Goals

Our intended user is anyone who needs to go to a store or various stores to do their shopping. The system is aimed to recommend a shopping route for the user to follow, based on the items they require. This route could be the shortest path for the minimal total expenses or the shortest travel time to purchase all the items.

2.1 Key High Level Goals and Problems

The user may not know where to go in order to purchase all the items they need. The user may also require a route that allows them to buy the items at the cheapest prices, or a route that takes the least amount of time. With this system, not only will the user be given a recommended route to shop for their items but it may also provide a route for minimal expenses or shortest travel time.

2.2 User Level Goals

To enter the items that they need to go shopping for and thereafter obtain a recommended shopping route to purchase these items for the minimal total expenses or the shortest travel time.

2.3 Environment

Either at home, work, or anywhere with online access.

3 Constraints

3.1 Legal Issues

We may come across issues where shops may not want to share their databases (such as items and prices) with us. Therefore, if we populate our system's database with such information, we may incur legal ramifications due to breach of confidentiality and incorrect listings of data.

3.2 Implementation and Software

If we do come across issues whereby we cannot use a shop's database, our system may not make use of it in the PathFinding Algorithm and therefore the shop won't be listed on the route. This could lead to a situation where the shop may be in the shortest path but because we cannot make use of it, the user may not obtain an accurate route based on their surroundings or shopping requirements.

We are limited by our implementation time as we are students with our own time constraints.

Since we plan to make use of Google Maps API for mapping the shopping routes, we are limited to the stores that exist on Google Maps.

Our system's database is limited to the regular pricing of items as we do not take into account item sales that can occur at shops. This also leads to a situation where our system generates a path that may not be accurate according to real-time changes.

The user is constrained to using a supported browser in order to make use of our web-application.

3.3 Hardware

Since our system is an online web-application, it is constrained by the response time and performance of the server and internet connection.

4 Pricing and Implementation Costs

The possible costs that we may incur are:

-A domain for the web-application.
R197 once-off (.co.za)
https://www.afrihost.com/site/product/domain_registration

-A server to host the web-application.
$358 upfront
(Standard 1 year term: t2.micro for Windows with SQL Web in EU-frankfurt)
https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/

-Development time and maintenance.
Subject to working hours and consulting rates of the development team.

5 Quality

5.1 Correctness

The user will create an account if they are not registered on the system. If the user has an account then they will log in. The user then enters the items they require and will run the SRRec when they want to go shopping.
When the user runs the software, the SRRec must take the user's list of items, compare it to the system's database, run the required PathFinding Algorithm (shortest path for the minimal total expenses or the shortest travel time) and return the mapped-out route to the user.

5.2 Reliability and Efficiency

Because the SRRec is a web-application, the reliability of the system is reliant on the performance and response time of the server and internet connection.

5.3 Integrity

Since the SRRec generates the required shortest paths by making use of the data from various shops' databases, we need to ensure that these databases are secure to protect each store's confidentiality.
We should make use of SSL, TSL, and HTTPS to ensure security within our system's online connections and data communications.

5.4 Usability and Portability

Since we do not plan on including a help function (for novice users) within our web-application, we want to implement the user interface to be as simple and user friendly as we can make it, to promote ease-of-use over ease-of-learning.
We will make use of HCI principles and keep consistency throughout the web-application's interfaces. This will allow users to become comfortable and familiar with using the system.
Since the SRRec is a online web-application, it can be used on all supporting web browsers (PC and mobile devices) and this allows the user to run the SRRec wherever they may be, granted they have good internet connection.

5.5 Maintainability, Flexibility and Reusability

Throughout implementation, we will develop the back-end and front-end software in a logical and consistent way so that we may easily make software changes if needed and if a problem occurs, we may locate and fix the issue easily. This also allows us to adequately reuse some parts of the software.

5.6 Testability

To ensure that our system is implemented correctly, we will make use of interface, validation and database testing.

Interface testing will be used to check that all UI components (such as buttons) work and trigger their respective processes such as passing data from the forms to the database, changing pages, etc.

Validation testing will be used to check and ensure that all data entered and passed from the interface forms to the database is of correct data type and format.

Database testing will be used to check and ensure that data is captured and stored correctly within their respective data stores and attributes. This also ensures that our PathFinding algorithm retrieves and makes use of correct data to provide an accurate route to the user.

Algorithm testing will be used to check and ensure that our PathFinding algorithm works and produces a correct route that is also mapped-out for the user.

6 External Interface Requirements

6.1 User Interfaces

The design of our user interface plans to include both elegance and functionality, the GUI design is planned to suit all users and allow for users of all ages to easily engage with the application. We plan to ensure that our application has ease of navigation and a user interface that is both stylish and current.

6.2 Hardware Interfaces

Due to the sheet scale of the project the need for dedicated devices and support hardware is not necessary at this point in time, but as the project start to gain popularity amongst users it could call for the use of large scale severs to sore all user information, that would need to be based around the world to ensure that all users from all corners of the globe get the same quality software.

6.3 Software Interfaces

Our software plans to make use of MVC(Model-View-Controller) architectural pattern which separates an application into three main components:
- Models : Models objects are the part of the application that implement the logic for the applications data domain, they often retrieve and store model state in a database.
- Views: Views are the components that display the applications user interface
- Controllers: Controllers are the components that handle user interaction, work with the model , and ultimately select a view to render that display. The controller handles and responds to user input and interaction.

Our database will be developed on a SQL server as it works well with the MVC framework as well as C-Sharp, has a huge amount of functionality and allows for scalability and growth of the software if needed.

7 System Features

7.1 Minimum Shopping Expenses 1

7.1.1 Description and Priority

Priority: Very High
The user inputs their shopping list of, say n, items the program then outputs a list of n or less (if 2 or more items are available at the same shop) shops that supply the items on the shopping list for the cheapest amount possible. The output will also show which shopping list items can be purchased at which shop. This feature gives the user no indication the order in which listed shops should be visited.

7.1.2 Stimulus and Response Sequence

-User types in shopping list
-User presses button labelled "TBD?"
-System outputs table of shops and products

7.1.3 Functional Requirements

The requirements for this feature are a database containing shops (for example: PNP,Checkers,game, DionWired, Mr Price, Woolworth) that stock items an average shopper may need, for example: clothing, groceries, appliances etc, and specific items stocked in said shops with their respective prices. The product should also respond to invalid inputs and errors with an error message. Invalid inputs include items entered that are not found in the database.

7.2 Minimum Shopping Expenses 2

7.2.1 Description and Priority

Priority: High
This feature is basically an extension of Minimum Shopping Expenses 1 (feature 2.1), but incorporates the vital goal of the product: a route to take. The user inputs their shopping list of, n, items and their current location. The program then outputs an **ordered list** of n or less shops to visit and which items are available to purchase at each shop. The shops listed are to provide the user the cheapest possible shopping expenses and are ordered to provide the shortest route while minimum price is prioritised. The route length is irrelevant.

Map: A further extension of this feature will be to show the shops on a map with directions of the route.

7.2.2 Stimulus and Response Sequence

-User types in shopping list
-User presses button labelled "?TBD?"
-System outputs table of shops and products
-System outputs map with directions

7.2.3 Functional Requirements

The basic requirements are the same as Minimum Shopping Expenses 1 with the added information of each shops GPS co-ordinates. As well as the added software capabilities of resolving precise locations and mapping.

7.3 Minimum Travel Time

7.3.1 Description and Priority

Priority: High
This feature uses the same input information as Minimum Shopping Expenses 2 (feature 2.2) (shopping list and current location) and outputs an ordered list of shops with preference to a short route length and travel time over actual shopping expenses.

Map: A further extension of this feature will be to show the shops on a map with directions of the route.

7.3.2 Stimulus and Response Sequence

-User types in shopping list
-User presses button labelled "?TBD?"
-System outputs table of shops and products
-System outputs map with directions

7.3.3 Functional Requirements

Will require software to calculate shortest route and perhaps traffic awareness and avoidance protocols.

7.4 Minimum Total Expenses

7.4.1 Description and Priority

Priority: High
This feature requires the same input as features 2.2 and 2.3 and outputs an ordered list of shops that takes into account both shopping expenses and travelling time and expenses and minimises both.

Map: A further extension of this feature will be to show the shops on a map with directions of the route.

7.4.2 Stimulus and Response Sequence

-User types in shopping list
-User presses button labelled "?TBD?"
-System outputs table of shops and products
-System outputs map with directions

7.4.3 Functional Requirements

This feature requires up-to-date information on travelling expenses (from the AAA or the likes). As well as the functional requirements listed in Features 2.2 and 2.3.

7.5 Account/Login

7.5.1 Description and Priority

Priority: Medium
This feature will enable users to sign-up and create an account which enables them to store pertinent information, such as:

- favourite shopping list
- favourite route or route type eg: Minimum Shopping Expenses
- home/work/starting location

for future use.
The inputs for this feature would be a valid user-name and password and some form of verification eg: email address or cellphone number.

7.5.2 Stimulus and Response Sequence

-User enters Sign-Up page to create an account
-System creates the necessary databases for the user
-User enters preferred information eg: favourite shopping list
-System saves information in correct tables in database

7.5.3 Functional Requirements

The requirements for this feature would extend to added databases to store user-name password keys, saved shopping lists and routes and saved locations. We would also have to provide security to ensure none of the users sensitive data is accessible.

7.6 Item Success Tracking

7.6.1 Description and Priority

Priority:Low
This feature would be an add-on to all the above route features and would remind you which items to purchase at the current shop and then track the success of your stop at the specific shop. This would enable you to recalculate your current route based on whether you found

the item at this shop or not. As well as notify other users of the availability of an item at a specific shop.

7.6.2 Stimulus and Response Sequence

-When the user arrives at a shop the system outputs which items off the users shopping list are intended to purchased at this shop.
-User inputs whether or not they were successful.

7.6.3 Functional Requirements

This feature would require constant access to the users GPS and perhaps some added input from the user.

8 Security Requirements

This product has several security and privacy issues that must be addressed. Concerning user authentication the product will require a valid email address to create an account, then they will require a unique user-name and password pair to ensure no cross contamination of private user information and to prevent access to the users sensitive and personal stored information, such as home or work address, email address, password and frequent or favourite routes. The developers will also have to provide guarantees to participating shops that provide classified and valuable databases of their stock prices that no malicious user or third party will be able to obtain this data beyond the intended scope of the product.

9 Team Contribution

Section 1: Storm Menges
Section 2: Sabeehah Ismail
Section 3: Sabeehah Ismail
Section 4: Sabeehah Ismail
Section 5: Sabeehah Ismail
Section 6: Sergio Oliviera
Section 7: Levi Goldfein
Section 8: Levi Goldfein