Schlep Application Software Requirements Documentation

Table of Contents

1. Introduction

1.1. Title: Schlep Application Software Requirements Documentation

1.1.1. Project Name: Schlep

1.1.2. Team Name: ThisGroupWillKillYourGroupNotSry

1.1.3. Date: 11/30/20

1.1.4. Team Members: Marc Bittle, Katelynn Urgitus, Josiah Stadler

1.1.5. UNCG Academic Integrity Policy

1.2. Table of Contents

1.3. Purpose

The purpose of the Schlep is to provide a type of ride sharing service that allows for delivery of large, or oversized products to be delivered to a customer when a seller does not provide delivery.

1.4. Document Conventions- 1.6 in example 1 srd

This document is organized as follows:

- Section 1 contains the introduction
- Section 2 provides a general description of the functionality and features of this project
- Section 3 details the functional requirements
- Section 4 describes the technical requirements
- Section 5 outlines the non-funcional requirements of this project

1.5. Intended Audience

The general public; specifically those with home improvement projects that require the hauling and delivery of items that require access to a pick up truck when a seller does not offer a delivery service, or the delivery fee is excessive.

Members of the public who purchase large items from sellers such as Facebook Marketplace and/or Craig's List without the means to transport these types of purchases.

Ike Quigley - Instructor of CSC-340 at UNCG

1.6. Definitions/Jargon

Schlepper - User of application registered as a driver, who facilitates pick up and delivery of items, and who may assist in the unloading process.

User/ Customer - User of application who utilizes the application to facilitate the transportation of an item

Schlep - The act of utilizing this application to physically transport an item.

Bid - Monetary and logistical information details pertaining to a specific instance of pick up, transportation and delivery.

Pick up location - GPS coordinates of location where item is to be collected by Schlepper.

Delivery location - GPS coordinates of location where item is to be off loaded.

Timer - Used to verify additional assistance time of driver

Additional items

1.7. Project Scope

The scope of the presented project contains all of the functionality for operations and customers as listed thus far. A finalized version of this project would include functionality for driver needs and information that would be required to facilitate a delivery. A finalized version would also include the ability to facilitate large organizations, already in the transportation and logistics sector, to utilize their current staff to increase revenue while adhering to Schlep guidelines and standards.

1.8. Technical Challenges

This application is intended to be used as a mobile device. Due to the restraints presented in this course, the application we are presenting will be desktop based and the mobile aspects of the application will be simulated.

Given the intended use of this application, it should feature real time GPS tracking data from pick up location to delivery location. The APIs required for this have a related cost. Since the development of this application is being done by college students, we elected to minimize group members expenses during its development. The GPS tracking elements of the presented application will simulate the functionality of the intended application without the inclusion of the aforementioned APIs.

2. Overall Description

2.1. Product Features

Signup Page

2.1.1.1. Customer

Login Page

Personal information

Delivery selection options

Additional needs requests for items essential to offloading

Payment information input page

Bid List Screen - includes Schlepper Acceptance Confirmation

Schlepper Current Location Tracking Screen

Timer Function - for assistance wage verification

Customer Wage Timer Function

2.2. User Characteristics

Users

2.2.1.1. Individuals with a need for a large item delivery who do not have a means of transporting said item.

Drivers (Schlepper)

2.2.1.2. Individuals who are employed by Schlep Inc. These individuals own a large vehicle such as a: pick up truck, box truck, flatbed truck and are paid for services rendered per the employee agreement.

Commercial Transportation Operation Owners

2.2.1.3. Owners of transportation services companies who wish to utilize the Schlep application to expand their revenue stream by using the company's current infrastructure. (To be implemented in future updates)

Drivers employed by a third party company (Super Schleppers)

2.2.1.4. Individuals currently employed by a commercial operation that supplies transportation solutions and has registered the operation with Schlep Inc. These companies can allow their current employees to utilize Schlep to transport goods as a standard Schlepper would using the company's existing resources such as vehicles, dollies, hand trucks, ETC. The pay scale and insurance requirements are reflected in the Commercial Operators Policy Document and are different from agreements pertaining to standard Schleppers. (To be implemented in future updates)

2.3. Operating Environment

Desktop

2.3.1.1. The application presented will operate on a desktop environment.

Mobile Device

2.3.1.2. The optimized and final version of this application is intended to operate on mobile devices. Due to the assignment requirements for CSC-340, the presented application will simulate this type of operation.

Residential Users

2.3.1.3. Private citizens, within a territory in which Schlep Inc. currently operates and can provide resources, with needs as described in section 2.2.1.1.

Residential Drivers (Schleppers)

2.3.1.4. Private citizens within a territory in which Schlep Inc. currently operates and can provide resources, who own resources and wish to earn wages through employment with Schlep Inc. as described in section 2.2.2.1.

Commercial Transportation Operation Owners

2.3.1.5. Owners of commercial operations who wish to expand their revenue stream by utilizing the Schlep application through the current company infrastructure, as described in section 2.2.3.1.

Employees of Commercial Transportation Operations

2.3.1.6. Individuals employed by a third party company currently registered with Schlep Inc. as Super Schleppers. These individuals will utilize the application per the operation's manager in accordance with the Commercial Operators Policy Document as described in section 2.2.4.1.

2.4. Design Implementation Constraints

Unable to access necessary API for real-time GPS tracking of drivers.
Will not be operational on mobile devices due to assignment specifications for CSC-340.
Processing of credit cards for payment will be simulated.
Driver information will be hardcoded/simulated due to aforementioned API and time constraints.

2.5. Assumptions and Dependencies

2.5.1. Assuming that the JDK's following JDK 11 do not have significant differences.

3. Functional Requirements

3.1. Primary

Contact Schlepper
Get Schlepper bids
Track delivery progress
Evaluate Schlepper assistance time for wages
Process payments
Delivery timeframe estimation

3.2. Secondary

Secure login
Technical support

4. Technical Requirements

4.1. Operating Systems/Compatibility

This application will primarily be used on the Windows operating system version 10 or higher.
This application will function on any platform that supports Java JDK 1.80_111.

4.2. Interface Requirements

Users will interact with the application through standard input and output devices such as a keyboard and mouse.

4.2.1.1. The ultimate goal is for users to be able to interact with the application through a mobile device.

There are no additional hardware requirements for the presented version of this application. However as stated above, the final version will require the use of a mobile device therefore it would need to access the GPS system hardware of that device. Ideally the application will support usage from both Android and IOS devices.
The version of the application being presented here connects to an API for the purpose of generating string values for use in the selection process of a user's address attributes.

4.2.1.2. Future versions will connect to a GPS API that facilitates the calculation of distance between: drivers' current locations and a pickup location, a pickup location and a user's address, and will

modify these distances in real time. This has been omitted from the presented version of the application due to the monetary requirements necessary to utilize such an API.

4.2.1.3. Future versions of this application will also connect to a POS API for the purpose of processing credit card transactions. This has been omitted from this version of the application due to the monetary requirements necessary to connect to such an API.

5. **Nonfunctional Requirements**
   - 5.1. Performance requirements
   - 5.2. Safety Recovery Requirements
   - 5.3. Security Requirements
     - 5.3.1. Secure login and signup page to protect personal data
     - 5.3.2. Secure processing of credit card payments through a trusted payment processing API. (Simulated in the presented version)
   - 5.4. Policy Requirements
     - 5.4.1. The presented desktop version of this application can be used in conjunction with third party applications designed for the purpose of supporting ADA compliance.
     - 5.4.2. The mobile version of this application (not presented here) will support third party applications designed for use on mobile devices to support ADA compliance.
     - 5.4.3. The presented version of this application is available in English only at the time of this presentation.
   - 5.5. Software Quality Attributes
     - 5.5.1. Availability
       - 5.5.1.1. From the user perspective, this application can be used to facilitate a same-day delivery from 10:00 a.m. to 12:00 p.m. and from 1:00 p.m. to 11:00 p.m. local time. When a user wishes to schedule a delivery in advance, they may do so at any time, but physical delivery will occur during the hours listed for delivery included in the instant delivery time windows. Meaning, A user can input a request for delivery at any time. However, in consideration of Schlepper safety and estimated rate of application usage, and estimated times of use, Schlepper activity (the process of a driver picking up an item and delivering it at a user's location) will be limited to the hours of 10:00 a.m. and 11:00 p.m. (local time). Furthermore a request entered by a user will not be processed, nor will bids be returned to the requesting user outside of these hours.
       - 5.5.1.2. Updates will be pushed to the application during inactive hours outlined above.
     - 5.5.2. Correctness

5.5.2.1. The majority of correctness requirements for the use of this application is related to the connections and use of specific APIs such as the CountryStateCity API which is used to populate dropdown lists that contain names supported regions for the purpose of optimizing performance where such values are needed for processing.

5.5.2.2. Cost of specific Schleps are calculated correctly prior to payments being processed. This is supported in multiple classes using the PassCost Model class.

5.5.2.3. Schlepper wages are calculated correctly and verified by the use of the wage timer when Schlepper assistance is needed.

5.5.3. Maintainability

5.5.3.1. The structure of this project is maintained in the MVC Architecture. This pattern ensures that related functionality and data is compartimentilized within corresponding files. More specifically, these files are segmented into folders categorized as models, views, controllers, where: files in the model folder contain business logic, files in the view folder contain implementation needed for various states and uses for user interactions, and files in the controller folder contain the logic to route appropriate data between the model and view files, as well as logic needed to coordinate interaction between outside resources. Outside resources include the API, and database.

5.5.4. Reusability

5.5.4.1. This project includes the use of external components (a Database and an API connection) which are vital to the functionality of the Schlep application. It is important to make use of good engineering practices in case of the event where we decide to use a different API or use a different Database. Building a Database interface to implement CRUD operations will allow for the use of any database . This makes the project data is not dependent on any one database and ensures that making such a change will not affect the functionality of the application.

5.5.5. Portability

5.5.5.1. This application is compatible with any platform that supports Java FX 15 and JDK 11.

5.6. Process Requirements

5.6.1. Development Processed Used

5.6.1.1. The MVC Architecture was used for this project to ensure that any business related logic, such as calculating the cost for a specific delivery, are kept separate from the user interface and any external components. In this Architecture there are three main components: Models, Views, and Controllers. The Models includes

all data and its related logic. The Views can present the data to the user or handle user interaction. The Controllers are the interface between the Models and Views. The Controller alerts the Model of any action taken by the user on the View. An example of this in Schlep is calculation of cost for delivery where we have a View that allows the user to select a location in which they need their item picked up and a location for where they need their item dropped off. They may also select whether they need the driver to provide additional items, such as a handtruck, or if they need physical assistance from the driver. The Controller then takes all of this data and passes to our Model that calculates the cost based on the user input.

5.6.2. Time Constraints

5.6.2.1. Total time given to work on this project was the entirety of the Fall 2020 semester which is sixteen weeks. As a group, the team spent approximately two and a half hours working together every Friday throughout the given timeframe. Individually, we spent between 6 and 40 hours per week on individual tasks. Due to these time constraints, the project does not include the functionality for a Schlep Driver.

5.6.3. Cost and Delivery Date

5.6.3.1. In terms of cost, we are currently using a free API connection for the purposes of demonstration. However, in a real world project we would need to consider the cost of maintaining an API connection. Other costs to consider are wages for developers, advertisement, and, when the application is completed, the wages for the drivers. The scheduled delivery of this project is for Tuesday December 2nd of 2020 at 5:30pm.