Concept of Operations

**Table of Contents**

1. Introduction

1.1 Purpose of Document

1.2 Contributors

1.3 Glossary of Terms

2. Scope

2.1 Version

2.2 Vision

2.3 Overview

2.3.1 Goal

2.3.1.1 Ride Requesters

2.3.1.2 Dispatchers

2.4) Development time frame

2.4.1) Timeline

2.5) Minimum Viable Product: basic system requirements

2.5.1) Front-end

2.5.1.1) Form

2.5.1.2) Dispatcher Schedule

2.5.1.3) Languages

2.5.2) Back-end

2.5.2.1) Languages

3. Current System

4. Nature of Changes to be made

4.1)Ride Request Form

4.2) Dispatcher Schedule

5. Responsibilities

5.1) Nicole

5.2) Jenny

5.3) Julie

7. Operational Scenarios

6.1) Typical Scenarios:

6.1.1) Ride Requester

6.1.2) Dispatcher

6.2) Extreme and Stress Scenarios:

6.2.1) Ride Requester

6.2.2) Dispatcher

**1. Introduction**

1.1) Purpose of Document

This document is intended to outline the basic concept for the SafeRide web application to be used by SafeRide (defined in section 1.3.1) in Eugene, Oregon. It is intended for users in order to clarify the reason, purpose, goals, and techniques involved in developing the SafeRide application. It will discuss the scope and a general overview of the project, the state of the current system and a justification for the changes to be made, basic concepts for the proposed system, operational scenarios, the impacts of the new system, and a final analysis.

1.2) Contributors

Contributors to the development of the SafeRide app will include:

Jenny Horn

Nicole Hsieh

Julie Stringham

1.3) Glossary of Terms:

**SafeRide** – a free shuttle service that provides university students, faculty and staff with an alternative to walking alone at night

**Dispatchers** – the SafeRide employees and volunteers who work at SafeRide headquarters and accept calls, schedule rides, and communicate ride requests with the drivers

**Ride** **Requesters/Riders** – users not affiliated with SafeRide who will use the application to request a ride from SafeRide

**Ride Request Form** – the user form which contains input fields in which Ride Requesters may input their information in order to request a ride

**Dispatcher Schedule** – the table, accessible only by dispatchers with log-in capabilities, which displays Ride Requesters’ information

**2) Scope of Project**

2.1) Version

This is the first version for the system. Version 1.0 will be a minimum viable product.

2.2) Vision

The vision for the system is to have an app that allows users to request rides easily and more efficiently by filling out an online form rather than having to call SafeRide headquarters.

2.3) Overview

2.3.1) Goal:

The goal of this project is to create a web-based application to automate the process of requesting rides from SafeRide.

2.3.1.1) Ride Requesters

Users will be able to fill in the information regarding their ride request into an online form and submit it instead of calling SafeRide on the phone. This will allow for easier access to SafeRide for all students, decrease the number of miscommunications caused by not being able to hear correctly over the phone, and save time for the SafeRide dispatchers.

2.3.1.2) Dispatchers

Dispatchers will be able to log in to access a table displaying the ride requests. The table will display all of the necessary information, have the ability to edit, add, or delete ride requests, and automatically show whether a requested location is in-bounds.

2.4) Development time frame: 4 weeks

2.4.1) Timeline:

Week 1: Project plan and design

Week 2: Basic Front-end form complete

Week 3: Client-Server connection, Dispatcher Schedule

Week 4: Location validation, details and extra features

2.5) Minimum Viable Product: basic system requirements

2.5.1) Front-end

2.5.1.1) Form

* + - * 1. Name
        2. UO Student ID
        3. Phone number
        4. Pick-up time
        5. Pick-up and drop-off locations
        6. Party size
        7. Comments
        8. Submit button at bottom

2.5.1.2) Dispatcher Schedule:

* + - * 1. Display requested rides
        2. Green if in-bounds, red if out-of-bounds
        3. Add ride request
        4. Delete ride request
        5. Modify ride request
        6. Clear all ride requests

2.5.1.3)Languages: HTML, JavaScript, CSS, bootstrap, Ajax

2.5.2) Back-end:

2.5.2.1) Languages: PHP & SQL

**3. Current System**

The current system in place for placing requests and dispatching vans at SafeRide involves no automation. Ride requesters must call SafeRide headquarters on the phone to make a request, and dispatchers at SafeRide enter the riders’ information into a Google spreadsheet by hand. Requests that are made before the headquarters’ operational hours are made by ride requesters leaving a voice message, which dispatchers later go through. When a request cannot be honored, a dispatcher calls the ride requester to inform them and reschedule.

**4. Nature of Changes to be made**

4.1)Rider application

- Riders will be able to enter the information about their ride into an online form in order to submit a request for a ride. The information includes the following:

* + - 1. Name
      2. University of Oregon Student ID number
      3. Email
      4. Phone number
      5. Desired pick-up time
      6. Number of members in the rider’s party
      7. Pick-up location
      8. Drop-off location
      9. Comments

- The current system for invalid requests will remain: if a request cannot be honored, dispatchers will still call the rider to reschedule their ride.

4.2) Dispatcher Application

- SafeRide dispatchers will have access to a table which displays the database of ride requests.

- Log-in information (username and password) will be required in order to access this page

- Each request will be displayed green if it is within the location bounds defined on the SafeRide website and red if it is out of bounds. Dispatchers will be able to decide whether or not to honor requests that are out-of-bounds.

- Dispatchers will be able to edit the information in a ride request for clarity, as well as add new rides requested via telephone.

**5. Responsibilities**

5.1) Nicole: Dispatcher Schedule, Database

5.2) Jenny: Ride Request Form, Documentation

5.3) Julie: Connection, location validation

**6. Operational Scenarios**

6.1) Typical Scenarios:

6.1.1) Ride Requester

A ride requester will access the Ride Request Form page. He or she will fill out their information accurately, select a desirable time to receive a ride, and select pick-up and drop-off locations that are within SafeRide-defined bounds. Then they will submit the request and wait. If they don’t get a call back from SafeRide dispatchers, their ride has been successfully scheduled.

6.1.2) Dispatcher

A dispatcher will log into the dispatcher page to view the Dispatcher Schedule. There they will find a table displaying all the rides that have been requested so far. If a ride is not going to work, either because of conflicting times, an out-of-bounds location, or invalid information, they will call or email the user with the phone number or email provided in order to cancel or reschedule the ride.

6.2) Extreme and Stress Scenarios:

6.2.1) Ride Requester

A ride requester may type in invalid information. The proposed system will validate locations and provide auto-complete capabilities for user convenience, will limit pick-up time options with drop-down menus, will make sure that the UOID and phone number sections have number inputs, and will make sure all areas of the form have been completed. It will not be able to check to make sure that the UOID is valid and active, or that the phone number, email address, name, or other fields are valid/accurate entries.

6.2.2) Dispatcher

A dispatcher can delete entries, and then will not have access to the contact information of the ride requesters in order to contact them to cancel their rides.