Project Plan:

Abstract

List of Tables

List of Figures

1. Introduction

1.1 Overview

1.2 Report Structure

1.3 Aims and Objectives

2. Literature review

3. Design & Methodology

3.1 Overview

3.2 Functional Requirements

3.3 Non-Functional Requirements

3.4 Clustering

3.5 Path Finding

3.6 GUI

3.7 System Design

4. Implementation

4.1 GUI

4.2 Clustering

4.3 Weather Data

4.4 Flight Path Correction

4.5 Map Drawing

4.6 Database Management

4.7 Path-Finding

4.8 HTML Output

5. Testing

5.1 Acceptance Testing

5.2 Performance Testing

5.3 Path Finding Algorithms

5.4 User Testing

Abstract

Introduction

Drones are an emerging technology for delivery. Many companies are developing systems to utilise them, such as Amazon with their “Prime Air” platform, and DHL with their “Parcelcopter”. They hold many benefits over traditional road-based delivery methods, such as lower cost, faster delivery and lower environmental impact. However, there are some obstacles to overcome before they see widespread use, such as legal issues and low flight range.

As with any delivery method, it is important that drones do not waste time. Their routes need to be optimised in order to speed up delivery for customers and companies alike. Because of this, it is vital that a schedule is created and maintained that details where each drone will be going and when.

Aims and Objectives

The aim of this project is to create a system to take input of customers and their locations, and create an efficient schedule for these customers. The following objectives must be completed in order to succeed at the task:

- Implement a simple Graphic User Interface (GUI)

- Develop a system to generate routes between customers

- Present the results graphically

- Allow comparison of different algorithms

Structure of Report

Chapter 2 – Literature Review

Chapter 3 – Design and Methodology to be implemented

Chapter 4 – Implementation of said design

Chapter 5 – Testing

Chapter 6 – Evaluation

Chapter 7 – Conclusions and future work

Literature Review

bla bla bla bla bla

Design and Methodology

This chapter will discuss the requirements of the project. The requirements are a direct result of the aims and objectives, and are steps required to complete these.

The methodology taken to implement the requirements is then explored.

Requirements

The requirements have been split into categories depending on importance using the MoSCoW system. Each requirement is classed as either “Must Have”, “Should Have”, “Could Have”, “Won’t Have”. For this project, “Won’t Have” has not been detailed.

“Must Have” requirements are those that the project cannot go without.

“Should Have” are important requirements, but not completely essential to the project.

“Could Have” are additional features that would be nice to have, but are a bonus, and their omission doesn’t affect the project in any way.

Functional Requirements

The functional requirements are functions of the implemented system. These detail tasks that the completed system will complete.

Must Have:

- A GUI – It is crucial to make the application usable to anyone that it has a simple GUI. This GUI has several characteristics that must be implemented in order to complete this task:

- A real map of a city. It allows the user to test out and see real-world results of their use.

- Input orders from the GUI. It is no good having a GUI if the user still has to input data using some other means

- Controllable parameters. One of the main aims of the project was to allow comparison of parameters. It will help users to see how adding or taking away drones for example will affect the results.

- Division of customers into smaller groups. As discussed in the Literature Review, it is vital that customers are broken down into manageable groups. Without this feature, the software will likely take far too long to complete path-finding, will be less likely to produce a good route, and also will not create a realistic scenario of multiple drones being able to fly at once.

- Sensible routes must be created. This is perhaps the main function of the application, and without it there is little to show. Routes must be created among customer groups that are sensible and follow some kind of rules in an attempt to optimise them.

Should Have:

- Multiple path finding algorithms implemented. This falls under the aim of comparing input parameters. It will be interesting for users to see and compare different algorithms, however providing there are other parameters that can be controlled, and one path-finding algorithm is included, this is not vital functionality.

- Consider weather conditions. One of the big drawbacks of drones is that they are greatly affected by the weather. In order to simulate a real-world system more accurately, this should be taken into consideration. Weather data such as wind speed, direction and precipitation may all be considered.

- Interactive user input. In a real-world system, the application would be running constantly and waiting for new orders to come in. If a new order is placed, the system shouldn’t shut down while it deals with this, it should be able to change an existing route or create a new one to accommodate the new customer.

Could Have:

- A display of the current state of routes as they are created. This would be an interesting feature to allow the user to see how the path-finding algorithm selected works.

- When running, show the location of each drone as they move through their route. This would be a bonus feature where the system is running in real time and updates with progress of drones along their routes.

Non-Functional Requirements

Non-Functional Requirements detail