

|  |  |
| --- | --- |
| **Name** | **Student Number** |
| Stephen Gaffney |  |
| Kerrie Lowe |  |
| Stephen Keenan | 14398016 |

COMP30830

Software Engineering

**Requirements:**

To develop a Web Application to Display occupancy and weather information for Dublin Bikes

**Client**:

Dublin Bikes

Table of Contents

[Introduction 3](#_Toc1298085)

[Purpose 3](#_Toc1298086)

[Scope 3](#_Toc1298087)

[Definitions 3](#_Toc1298088)

[Overview 3](#_Toc1298089)

[Requirements Analysis Summary 3](#_Toc1298090)

[Data Collecting 3](#_Toc1298091)

[**Pitch / Brief Analysis** 3](#_Toc1298092)

[**System Observation** 3](#_Toc1298093)

[**Brainstorming** 4](#_Toc1298094)

[Requirements Analysis 4](#_Toc1298095)

[Selecting Data 4](#_Toc1298096)

[Analysing Data 4](#_Toc1298097)

[Overall Description 5](#_Toc1298098)

[Product Overview 5](#_Toc1298099)

[**System Interfaces** 5](#_Toc1298100)

[**Interfaces** 5](#_Toc1298101)

[**Hardware Interfaces** 5](#_Toc1298102)

[**Software Interfaces** 5](#_Toc1298103)

[**Memory Constraints** 5](#_Toc1298104)

[Product Functions 5](#_Toc1298105)

[User Characteristics 5](#_Toc1298106)

[Constraints 5](#_Toc1298107)

[Assumptions & Dependencies 5](#_Toc1298108)

[Appointment of Requirements 5](#_Toc1298109)

[Requirements Statement 5](#_Toc1298110)

[User Requirements 5](#_Toc1298111)

[Functional Requirements 7](#_Toc1298112)

[Non-Functional Requirements 9](#_Toc1298113)

[Performance Requirements 9](#_Toc1298114)

[Domain Requirements 10](#_Toc1298115)

[Design Requirements 11](#_Toc1298116)

[Change Management Process 11](#_Toc1298117)

[Approval 12](#_Toc1298118)

[Appendix 12](#_Toc1298119)

­­

# **Introduction – Anyone**

## Purpose

## Scope

## Definitions

## Overview

# **Requirements Analysis Summary – Stephen K**

## Data Collecting

The following approach was taken during the requirements elicitation process. Requirements were elicited through (1) System Observation, (2) Customer Interview (3) and Brainstorming.

### **System Observation**

The first method of elicitation was the observation of the current system. Through this, we aimed to have a greater understanding of who the system users were and how they interacted with the legacy systems. This ensured that we were not missing any pinch points.

Our research was focused on two main observations: general observation and the current DublinBikes web application. General observation was constructed from observations that we could make including: general experience; accuracy of information; and product offering. The DublinBikes application was reviewed by the team. We decided to borrow a common business practice called ‘SWOT analysis’.

### **Customer Interview**

The second method of elicitation was interviewing customers of the DublinBikes network and web application. We used people who we knew have used the service. The customers ranged in age and usage.

### **Brainstorming**

We believed that the technology that was required to produce this application was already been successfully implemented in a number of other industries. For this reason, we decided to research what was currently being offered individually and concluded by continuing the process as a team.

## Requirements Analysis

## **Selecting Data**

We were aware that there was a considerable amount of information collected as a result of the various elicitation techniques deployed. This required us to select the most relevant information that was applicable when determining the products requirements.

## **Analysing Data**

We chose to divide up the data that we gathered into clusters. We broke down each of the operations of the product into its own parts: (1) User Interaction / Information Display (2) Analytics (3) Display (4) Accuracy

The segregation allowed us to accurately identify what was needed. By dividing up the elements of the product, we see the features that should take priority. Using the information from the system observation, customer interview and brainstorming session, we outlined the features that were pinnacle to the success of the product.

* User Interaction / Information Display
* Analytics
* Display
* Accuracy

***Our requirements analysis highlighted the shortcomings of the current system. We believe that we have equipped ourselves well for producing a product that understands the needs of our customers. It is envisaged that the product will be fully integrated into the Moodle platform and that only necessary requirements will be included – see below.***

# **Overall Description – Stephen G**

## Product Overview

### **System Interfaces**

### **Interfaces**

### **Hardware Interfaces**

### **Software Interfaces**

### **Memory Constraints**

## Product Functions

## User Characteristics

## Constraints

## Assumptions & Dependencies

## Appointment of Requirements

# **Requirements Statement – Stephen K**

## User Requirements

The following are the features that the application will perform. This is an overall of an existing system, and as such is a considerable task to undertake. Therefore, and with the deadline in mind, we have decided to prioritise the features to be built. These features are listed under the headings: (1) Essential Features (2) Value-adding features (3) Non-Essential Features.

**Essential Features:**

1. A web application built using Flask
   1. Display info for drop off stations Map / Icon for full
   2. Display info for collect stations (Map, Icons for with and without
   3. Filter location (Eircode 500m radius)
   4. Display bike information for dropping and collecting on specific day – input box shall display amount on path history and weather)
   5. Display station info – number address available bikes and weather
2. A connection to the DublinBikes API
3. A connection to the OpenWeather API
4. A database to store information from Dublin Bikes API
5. A database to store information from OpenWeather API
6. A model to analysis the information from the DublinBikes API
7. Help section

**Value-Adding Features:**

1. Route planning
2. To estimate the time taken to cycle your route
3. Filter box for things to do – pull information from entertainment websites
4. Send reminder for your planned journey (15 mins before)

**Non-Essential Features:**

## Functional Requirements

## Non-Functional Requirements

## Performance Requirements

Performance Requirements are important in the development of any system as they provide a clear benchmark that the is expected from the performance of the product. This section will focus on how the system will perform under (1) *Response Time* (2) *Workload* (3) *Security* (4) *Accuracy.*

## Domain Requirements

1. **Security and Privacy**
   1. We assume that the browser is not compromised
   2. The Moodle login feature is functional
   3. The database will be in-line with GDPR Regulation (EU) (2016/679) and will hold only necessary information regarding members, staff and supplier contact information
   4. We assume students will be aware of the storing of information
   5. We assume professors will be aware of the storing of information
   6. We will duplicate all information in case of technical failures
2. **Storage**
   1. AWS will be flexible. Storage will be added as required.

## Design Requirements

* 1. Except for those cases where the user has been notified of data gathering, the plug-in should not gather data on the user
  2. All information that is gathered should be stored securely and deleted when deemed obsolete, or to be in compliance with GDPR standards.

# **Change Management Process**

The SRS will be modified as the requirements change. These changes can come about from (1) customer feedback (2) feature implantation issues (3) budget (4) external events. These changes will be logged and updated in a new version – will implement the industry standard of version control. These changes will be made by the Product Owner.

We agree that the team must reach an agreement to change the requirements. It is taken that the current issue of the document is, with the information that we have available, the best ‘action plan’ for producing the product. In order to make the changes, the team will sign off on the changes (those who it affects). This will stand as an agreement.

# **Approval**

The above SRS has been approved by the Product Owner ‘*Stephen Keenan’.* Any queries should be directed at the owner. All information contained in this document is original copyright of CS Technologies. The product developed should be done so for the named client.

*Stephen Gaffney Kerrie Lowe Stephen Keenan*

*20th April 2019 20th April 2019 20th April 2019*

**Title Title Title**

# **Appendix**

**Appendix 1 - User Story**

**Appendix 2 – RD Tables**

|  |  |  |
| --- | --- | --- |
| **Table Name: Student** | | |
| **Field Name** | **Caption** | **Data Type** |
| Student\_ID | Student ID | AutoNumber (Primary Key) |
| First\_Name | First Name | ShortText(30) |
| Last\_Name | Last Name | ShortText(30) |
| University | University | ShortText(30) |
| Submission | Submission | ShortText(20) |
| Submitted | Submitted | Bool() |
| Email | Contact Email | ShortText(20) |

|  |  |  |
| --- | --- | --- |
| **Table Name: Professor** | | |
| **Field Name** | **Caption** | **Data Type** |
| Professor\_ID | Professor ID | AutoNumber (Primary Key) |
| First\_Name | First Name | ShortText(30) |
| Last\_Name | Last Name | ShortText(30) |
| University | University | ShortText(30) |
| Solution | Solution | ShortText(20) |
| Submitted | Submitted | Bool() |
| Email | Contact Email | ShortText(20) |