**Software Requirements**

**Specification**

**for**

***Dublin Bikes***

**Prepared by *Mugesh*, Tehsein, Emma**

***UCD***

# 1. Introduction

## 1.1 Purpose:

The main objective of this document is to provide a comprehensive outline of the software requirements for the Dublin Bikes Web Application, aimed at providing details regarding bikes available at different locations across Dublin.

## 1.2 Scope

## The web application for Dublin Bikes enables users to check bike availability across the city via a map view. Users can also access more detailed information about bike availability at individual stations. Additionally, an ML model will be implemented to forecast station occupancy based on weather conditions.

# 2. Overall Description

## 2.1 Product Perspective

## The system is composed entirely of a web application that uses an ML algorithm and APIs to present information on the app. The system is designed for bike users and necessitates that users register for the service. The user information will be stored in a distinct database that will assist with authentication. All actions in the web application are carried out on the cloud.

## A crucial aspect of the web application is its ability to forecast occupancy using an ML Model. This entails scraping real-time data, and the database is trained using data collected over a duration of time.

## 2.2 Product Features

The system consists of various features which are given below:

1. To view the occupancy and availability of bikes in a map view
2. To view detailed information at each Bike stations.
3. To display weather information.

## 2.3 User Classes and Characteristics

The users will be required to sign in to access functions.

## 2.4 Operating Environment

The application will be designed to be compatible with any device with internet access. We anticipate that API requests / ML outputs will receive responses within 1 to 2 seconds.

## 2.5 Design and Implementation Constraints

## The efficiency of the ML model used to forecast station occupancy will be a critical constraint.

## Multiple users may access the system simultaneously, resulting in a high volume of requests being sent simultaneously. This may cause response delays.

## 2.6 Assumptions and Dependencies

1. It is assumed that the users use the web application with latest devices.

# 4. External Interface Requirements

## 4.1 User Interfaces

The chosen user interface has a minimalist design to ensure ease of use for the users. The majority of functions can be accessed through small buttons on menu, accompanied by straightforward webpages with text fields.

## 4.2 Hardware Interfaces

Since the web application have any designated hardware, it does not have any direct hardware interfaces.

## 4.3 Software Interfaces

ML model is being used to display occupancies and a database will be used to authenticate users.

## 4.2 Data fields from JcDeaux to be used:

1. Number
2. Name
3. Address
4. Position – Latitude and Longitude
5. Availability
6. Capacity
7. Overflow

### 4.4 Weather API

We’ll be using either meteostat python library or meteostat JSON API whichever convenient.

1. Temperature
2. Wind speed
3. Dew point
4. Humidity
5. Min temp
6. Max temp
7. feels like

## 4.5 ML Models in our mind.

1. Linear Regression
2. Catboost
3. XGBoost
4. LightBoost