SOFTWARE REQUIREMENTS FOR THE DUBLIN BIKE WEBSITE

Assignment 4

Software Engineering
COMP30670
4/20/2017

Emma Byrne – Katherine Campbell – Pamela Kelly

Contents

1. Introduction

1.1 Purpose

This document outlines and describes the requirement specifications for the Dublin Bikes website (herein DublinBikes.com), a website, tracker, and resource which assists cyclists availing the Dublin bikes system by showing and mapping all possible bike stations, indicating how many bikes and stands are at each station at the current time, supplying statistics such as average number of bikes and stands available for a particular station on a particular day and time, indicating whether or not a stand takes a credit card (or banking available) and notifying them of the current weather. This document provides an over-arching framework for (1) the tasks DublinBikes.com undertakes and achieves, (2) the scope of the website, (3) the user input – drop down menus to see detailed statistics, (4) the organization of the output – station locations, weather, statistics, etc., and (5) the beneficial and useful services DublinBikes.com provides. This is version 0.1 of these software requirement specifications.

1.2 Document Conventions and Terminology

This document employs and references the following terms and abbreviations:

- *Dublin Bikes*: A public bicycle rental scheme operating within the city of Dublin, Ireland
- *GUI*: Graphical User Interface
- *Google Charts*: a JavaScript based charting library from Google which adds interactive charting capability to webpages (Tutorialspoint ,2017).
- *DB*: Database
- *Flask*: a micro web framework for Python
- *SQLalchemy*: A Python library which allows users to reference and interface with relational databases
- *API*: Application Programming Interface – allows a programmer to create apps that use features or data of a specific service, app, or system (Gazarov, P. ,2016).

1.3. Intended Audience and Reading Suggestions

This document may be of use or interest to those in charge of or interested in the Dublin Bike scheme, users of said scheme, webpage designers, software developers and engineers, researchers and practitioners of both front and back end development (and those that link the two), data analysts, and business people interested in technological representation of large schemes or programs, among others. It is the hope that this document might shed some light on a functional way to quickly, cheaply, and effectively present and process a large amount of data in a comprehensive way for a layperson's use, possibly drawing attention to or sparking discussion regarding possible improvements or various uses of such a type of website.

1.4. Scope

The DublinBikes website is an application for users, and perhaps employees of Dublin Bikes. This website acts as a platform which provide cyclists with Dublin Bikes with vital information, such as available bikes and stands at a particular station, both current and at a given hour and day, a map with the location of each station, the current weather, and whether or not a stand takes credit cards as payment.

This tool facilitates and enhances a Dublin Bikes user's experience. It does not function as a platform for advertisement, Dublin Bikes membership sign up/termination/payment nor as a customer feedback site (although such things may be considered later and certainly feedback post-launch would absolutely shape the future of this site). Any information or services that lay outside of both current and average dynamic or static station information, current weather information, or user flow via heat maps lay outside the scope of this site.

This document does not cover API nor database design, nor hosting, merely noting the impact these aspects had on the product at hand. This SRS only outlines requirements for the application as an overview of whole-system functionality.

1.5. References

The format and outlining of this document was based on previously given examples such as *Tschau Sepp LOGIC Sub-Component: Software Requirements Specification* by Alexandru Dima, Olivier Clerc and Alejandro Garcia; *Software Requirements Specification for Internetworking of Content Delivery Networks through Peering Version 0.1* by Mukaddim Pathan; and *Software Requirements Specification for Project Management System project* by Ilyin Yevgenly, as well as our previously written SRS project.

For details outlining the proper definitions and uses of APIs and Google Docs, the following sources were consulted:

Gazarov, P. (2016, August13). What is an API? In English, please . Retrieved April 19, 2017, from https://medium.freecodecamp.com/what-is-an-api-in-english-please-b880a3214a82

Tutorialspoint. (2017). Google Charts Tutorial. Retrieved April 19, 2017, from http://www.tutorialspoint.com/googlecharts/

2. Overall Description

2.1. Product Perspective

Dublin Bikes website is a supplementary website to DublinBikes.ie which provides participants in this scheme with information regarding the weather, stations, and user flow. This site has only one page/interface – the one used by the cyclists. The information displayed on the site comes from the

Dublin Bikes API (for the station information) and OpenWeather API (for the weather information). A user need not be familiar with nor understand how to access or read said APIs as the site funnels, processes, and interprets the data for users.

2.2. Product Functions

- The functions of this application are:
- Providing users with a map of Dublin complete with markers for each station location
- Connecting to an amazon database (which is, in turn connected to the Dublin Bikes API getting the data via a scraper)
- Connecting to an amazon database (which is, in turn connected to the OpenWeather API getting the data via a scraper)
- Backing up the data from the database in local files in case of a db crash
- Revealing real-time information from the aforementioned API regarding individual stations (such as bikes and stands currently available)
- Connecting to the OpenWeather API and displaying the current weather (with appropriate icon), temperature, and windspeed
- Providing daily and hourly charts via Google Charts regarding average bikes and stands available at each station
- Providing daily and hourly charts as above but also in accordance with the weather (ex: average number of bikes on a Monday at 12pm when its sunny)
- Showing users whether or not banking is available at a particular station
- Showing users the present number of people in a location via a heat map
- All information, such as the queries to the database run through flask applications which then push them to the front page.

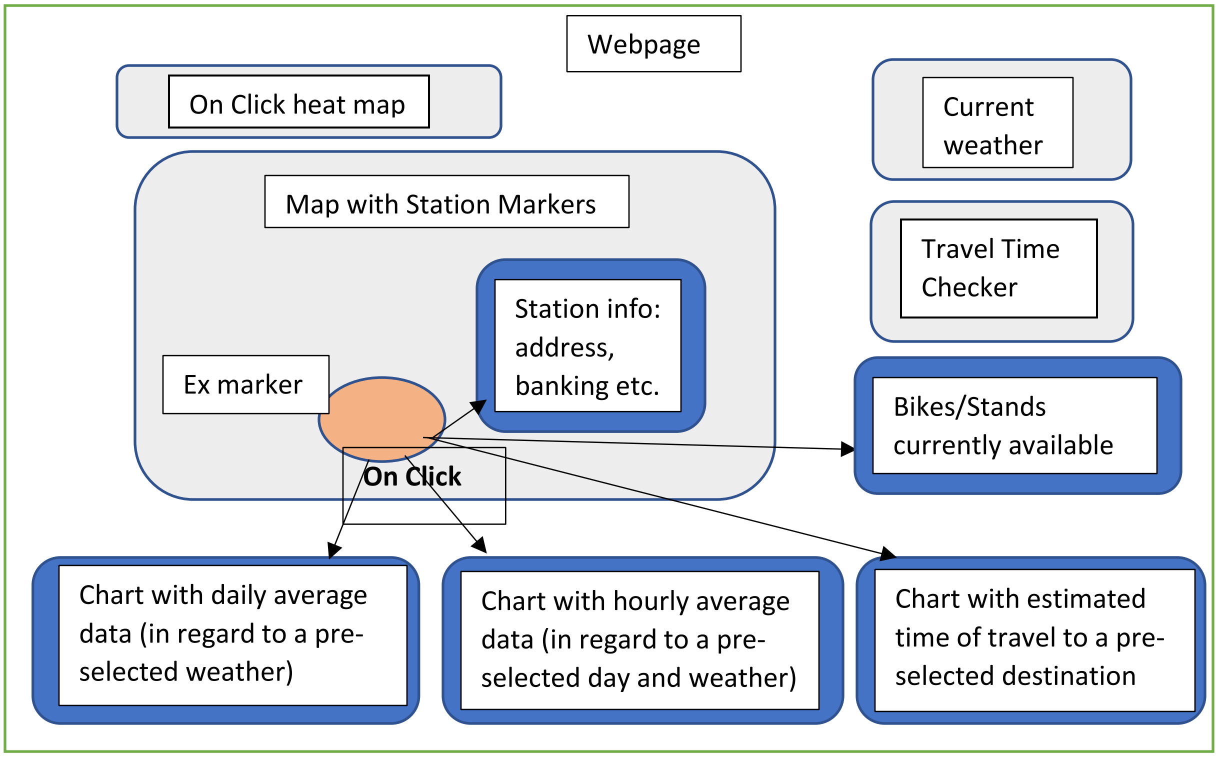
Webpage
On Click heat map
Current weather
Map with Station Markers
Travel Time Checker
Station info: address, banking etc.
Ex marker
Bikes/Stands currently available
On Click
Chart with daily average data (in regard to a pre-selected weather)
Chart with hourly average data (in regard to a pre-selected day and weather)
Chart with estimated time of travel to a pre-selected destination

Fig. 1: Summary of the Contents of the Webpage

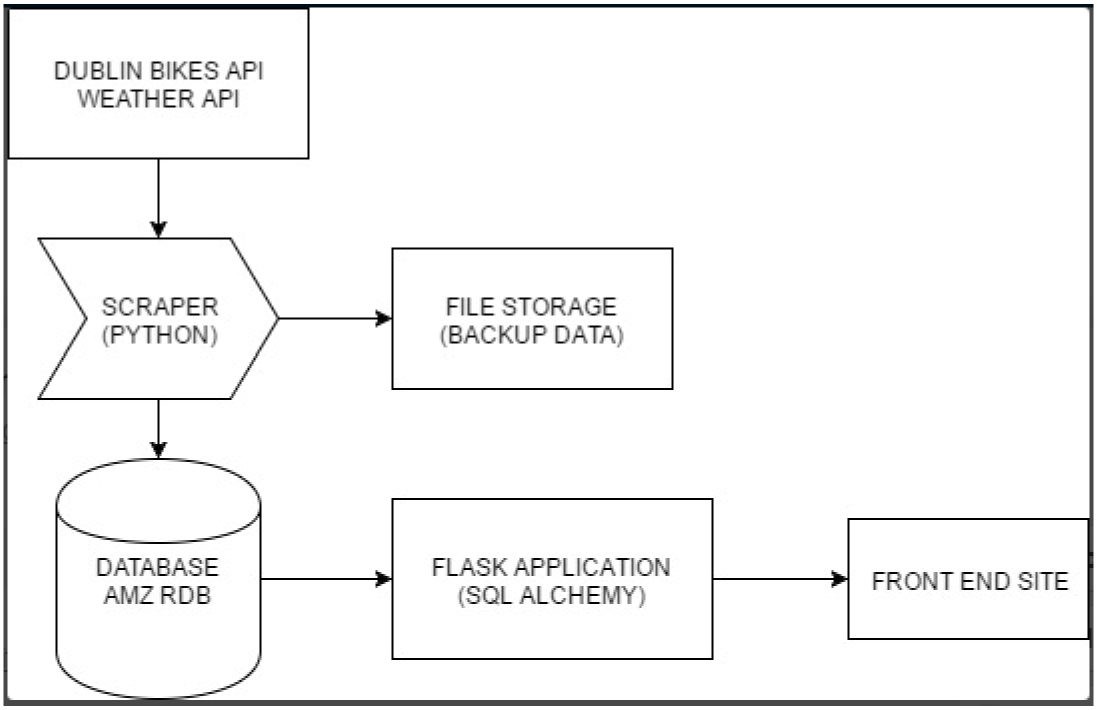

Fig.2: A rough overview of the interacting components providing data to the website

2.3. User Classes and Characteristics

There is only one type of user (although their reasons for wanting the information provided by the website may vary) – everyone has the same permissions and access to the same features. The user may access the website, and thus all of its features and functions simply by going to the appropriate URL.

2.4. Operating Environment

This website is responsive and can run on the most current versions of any internet browser as well as on Linux, Mac OS, Windows 7,8,10, android, and iOS operating systems.

2.5. User Documentation

A ReadMe will be provided for any user interested. This ReadMe will follow the typical format, outlining any packages that were used in the construction of this website, and highlighting key features and operations. Because most aspects of this website are intuitive and the behind-the-scenes construction are not of interest to the general public, a user-manual will not be provided.

2.7. Assumptions/Dependencies

It is assumed that all users have a functional and dependable internet connection.

3. System Features

3.1. Maps with Responsive Markers

3.1.1. Description

This feature shows the location of each Dublin Bike station. It consists of a Map (provided by Google Maps) of Dublin City Center. It has markers indicating each Dublin Bike Station. Each station, revealed upon clicking, has its own information box with the name, address, whether banking is available, and number. There is also a button to click for more information, revealing the charts outlined below.

3.1.2. Stimulus/Response Sequences

The map and markers load as the page loads. The information box for each marker only appears onclick.

3.2 Chart of Daily Averages

3.2.1. Description

All data regarding the bikes and stands available at a particular bike station have been scraped from the API for many weeks. Thus, a reliable average of these two features can be gleaned. This information has been put into Google Charts and a histogram with each day of the week appears revealing the average number of bikes and stands at that station. There are also more charts available with these averages but accounting for the weather (rainy/windy/etc.). This information is gathered from the database holding information being scraped from the OpenWeather API.

3.2.2. Stimulus/Response Sequences

The general chart appears upon the user clicking the 'more information' button on the original popup that appears when clicking a station marker. The weather specific charts appear when the user selects an option from a drop-down menu.

3.3 Chart of Hourly Averages

3.3.1. Description

As above, all data regarding the bikes and stands available at a particular bike station have been divided by day and time. This information has been put into Google Charts and a histogram with each hour of each day of the week appears revealing the average number of bikes and stands at that station at that particular hour on that particular day. There are also more charts available with these averages but accounting for the weather (rainy/windy/etc.). This information is gathered from the database holding information being scraped from the OpenWeather API. Users may select the day and weather.

3.3.2. Stimulus/Response Sequences

The hourly chart takes in the current day and appears with all the hours of the current day upon the user clicking the 'more information' button on the original popup that appears when clicking a station marker. The day can be changed via drop down menu and the related chart will appear. As the chart outlined previously, the same chart but accounting for different weather conditions exist and can be selected via drop-down menu.

3.4 Current Weather

3.4.1. Description

Data from the OpenWeather API is linked to a div in the website revealing the current weather, temperature, and windspeed, with corresponding icons.

3.4.2. Stimulus/Response Sequences

The above mentioned div appears when the page loads

3.5 Heat Map

3.5.1. Description

This heat map overlays the previously outlined map of all station markers. It shows how many people are at or between stations at the time of map loading.

3.5.2. Stimulus/Response Sequences

The heat map only appears once the user clicks a button to show it. Otherwise it is hidden. The data from the heatmap corresponds to the time in which the page was loaded.

3.6 Fun Facts

3.6.1. Description

A div on the page showing fun facts about cycling via a loop.

3.6.2. Stimulus/Response Sequences

The div appears upon the page loading. It loops through the facts showing a new one about every ten seconds.

3.7. Travel Time Checker

3.7.1. Description

A div on the page will allow the user to select two station names and it calculates and shows the estimated travel time between those two points.

3.7.2. Stimulus/Response Sequences

The div itself will load with the page, but the user must input the names of the stations via a drop-down menu and then click submit for the distance to be queried and then revealed.

3.8. Data to File Saver

3.8.1. Description

Because a database might crash, to ensure that the information provided to our users is up-to-date and doesn't also disappear, the scraper code retrieving the information from the respective APIs not only saves said info into the db but also in local files. Should anything go awry, the data still exists and the website still stands.

3.8.2. Stimulus/Response Sequences

This feature runs as long as the scraper is running. The local files will hopefully never have to be implemented but can be accessed at any time by those maintaining the site.

4. Other Non-functional Requirements

4.1. Security Requirement

Because the website doesn’t ask users for any identifying information whatsoever, the security from the user’s point of view is almost a moot point. The designers will never place any adds so pop-ups and cookies from other parties are not a concern. On the developer end of things, database passwords and API keys are stored in secure files that have not been shared at all and are absolutely necessary to run any code pertaining to said APIs and dbs.

4.2. Software Quality Attributes

- *Availability* – The program will always be available to whomever has access to the internet
- *Correctness* – All above outlined forms and user features will be accessed via the main webpage. These will be tested thus ensuring that they run as described.
- *Flexibility*- This app can be updated by the developers at any time
- *Security* – As mentioned above, the website does not ask for any identifying information. There are no adds and, on the developer side, all dbs and APIs need a password which are kept on individual computers in protected files
- *Maintainability*- This website will be designed for maintainability from the beginning. The code will be readable and refactored to improve understandability. Proper documentation (ReadMe) pertaining to common problems, as well as a list of all possible packages with their respective uses have been written and are available to every customer.
- *Portability*- The Dublin Bikes website will be available in Chrome, Firefox, IExplorer, as well as on the Linux, Mac OS, Windows 7, 8, 10, android, and iOS operating systems.
- *Reliability* – The information in the databases have been stored on local files in case of a db crash. The weather table and the Dublin bikes table with the pertinent data have been kept in separate dbs in the case of one crashing to ensure the other is not affected.
- *Robustness* – The data-from-scraper file backup system as outlined in sections 2.2 and 3.8
- *Usability* - The homepage will be intuitive with few and well-marked markers and buttons for more information and useful drop-down menus. There will also be tabs for home, about, and help/FAQs.