DublinBikes

System Requirement Specifications

Simplified Version 0.2

Software Engineering Project
Computer Science Conversion Program
University College Dublin

Group 9

Scarlet Grant
Jiarui Qin
Xiang Zhao

Changelog

Version	Date YY/MM/DD	Editor	Subject	Changes
01	x	Scarlet Grant		First concept Functional Requirements, delivered to the product owner
01	2019/03/28	Scarlet Grant	N/A	Added index and changelog
02	3/28/2019	Jiarui Qin	4.1	Use Balsamiq to change the user interface.
03	28/03/2019	Xiang Zhao		

1. Introduction

1.1 Purpose

Software requirements (Version 0.1) regarding the creation of a standalone app to help Dublin Bike users in their journey.

1.2 Product Scope

The app is created to remove the friction of biking in Dublin. The app aims to help Dublin Bike users with the following benefits as provided by the client:
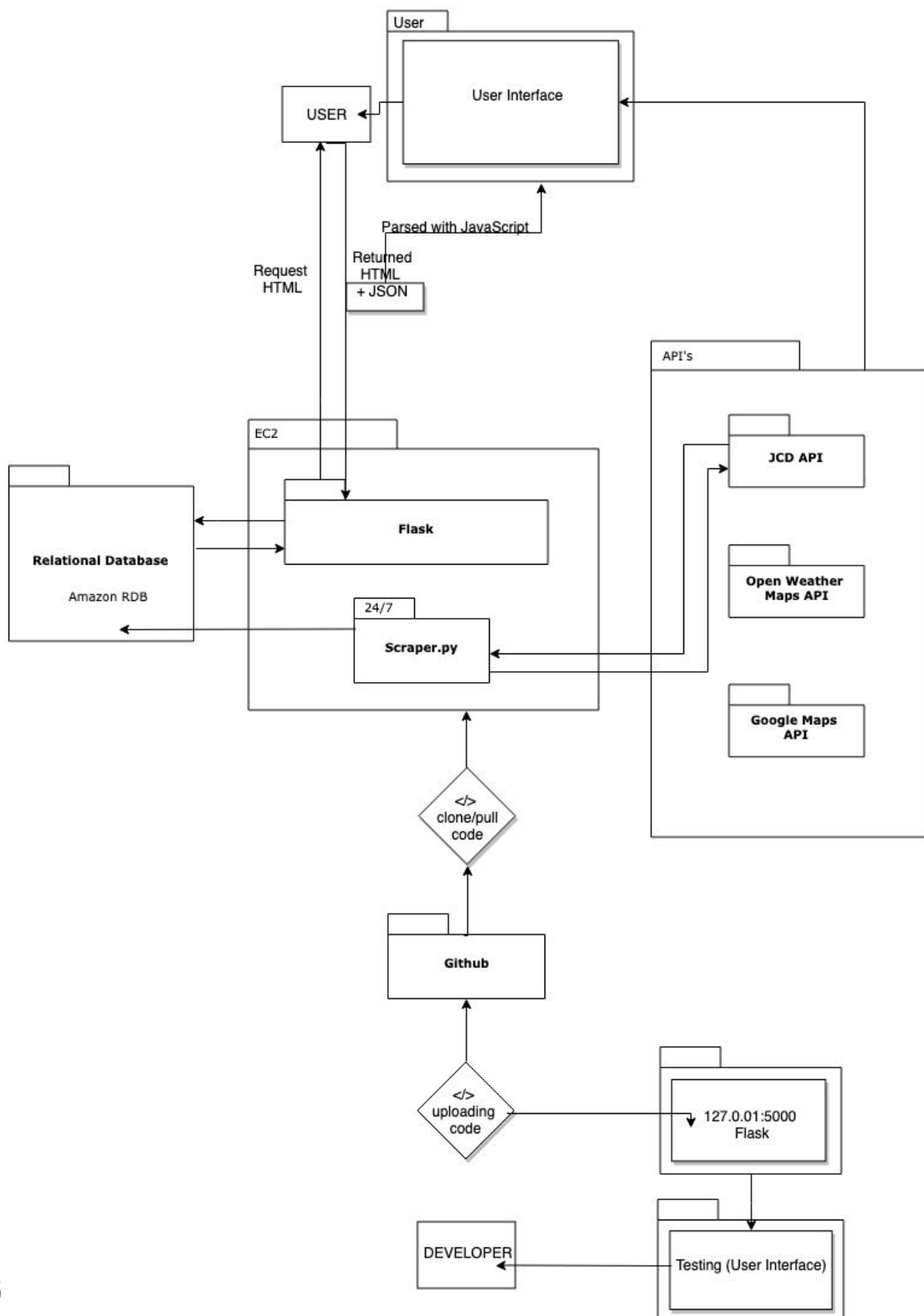
•Find the best station with available bikes to start a journey

•Find the best station with free spaces to return the bike

•Should display relevant and useful information learned from recent occupancy trends

•Incorporate weather information, as it affects the occupancy

•Include other sources of information

2. Overall description

2.1 Product Perspective

This stand alone app has no affiliations with dublinbikes.ie. The app makes use of JCDecaux bike information API and DarkSky API. See diagram below for an overview of the components and interconnections.

Basic Flowchart of app

User

User Interface

USER

Parsed with JavaScript

Request
HTML

Returned
HTML
+ JSON

API's

EC2

JCD API

Flask

Open Weather
Maps API

Relational Database

Amazon RDB

24/7

Scraper.py

Google Maps
API

</>
clone/pull
code

Github

</>
uploading
code

127.0.01:5000
Flask

DEVELOPER

Testing (User Interface)

5

2.2 Product Functions

- Show weather-forecast;
- Find nearest bike at desired/current location in least amount of clicks;
- Find nearest dock at desired/current destination in least amount of clicks;
- Show forecast of bike availability (at given datetime and location);
- Show availability for card payments at given location.

Approved adjustments of product briefing: Show weather upon opening the app

The project description of the client informs that the weather only needs to be shown after the after the station is clicked and shown. However, the team is of the opinion that the user wants to see the weather around him/her or in Dublin before it takes the effort to look up a specific Dublin station. The weather (and/or nearest station) should be visible on opening of the app. This adjustment has been proposed and approved by the Product Owner.

2.3 User classes and characteristics

The user is not expected to have any technical knowledge. The users might be once off users who try out Dublin Biking or they could be frequent users who will rely on the app. It is likely they will use the app mostly on mobile before starting their journey.

2.4 Operating environment

Taking the expected usage in consideration, the app will be firstly designed and build for mobile use, with a responsive scheme for larger screen usage. It also should be able to responsive under complex situation. It is supposed to work on any standard internet browser. Future applications could be developed for the Smart Watch and that would not be in the scope of this part of the project.

2.5 Contribution to Bike-user:

Offer convenience by friction less biking; Show weather-forecast (to bike or not to bike); Find nearest bike and dock at destination in least amount of clicks; show predicted bik availability at a given time.

2.6 Contribution to other stakeholders:

Dublin City: stimulate use of their biking scheme by eliminating friction

Advertisers: extra channel for green brand awareness

2.7 User stories

As a user, I can:

- open the app and see immediately the weather forecast that has me decide to bike or not to bike;
- open the app and see immediately the availability at a particular (preferably the closest) bike station;
- view the predicted bike availability at a particular data/time in the future;
- open the app and see quickly where I can return my bike;
- Open the application and find the nearest bike station.

3. System features

3.1 Assumptions

User uses mobile phone as primary device to open the app

3.2 Input

The bike information fetched from JCD;

The weather information fetched from Dark Sky API;

The geocoding information fetched form Mapbox;

The user current location;

user_choices-interaction-data

3.3 Output
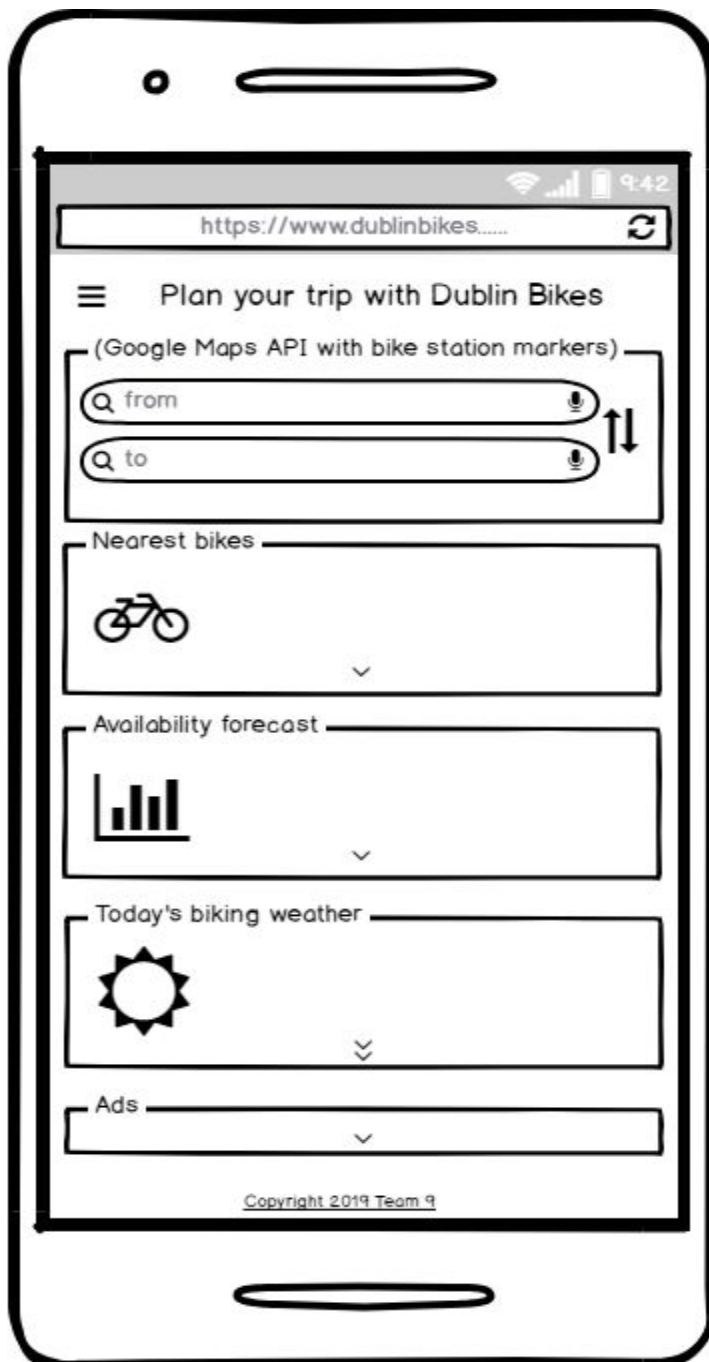
Nearest-available-bike-station&docking;

Predicted-bike-availability(based on historical bike-data, weather-history and weather forecast.

3.4 Overview functionalities **(needs further specification)**

- Show weather-forecast (to bike or not to bike);
- Find nearest bike at desired/current location in least amount of clicks;
- Find nearest dock at desired/current destination in least amount of clicks;
- Show forecast of bike availability (at given datetime and location);
- Show availability for card payments at given location.

4. Wireframes

4.1 Mobile

4.2 Desktop