Software Requirements Specification

for

<tweeLights>

Version 1.0 approved

Prepared by <Yasser El Kabbout>

<Bogazici University>

<10/3/2018>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Yasser El Kabbout	19/3/2018	Review – spelling	1.0

1. Introduction

1.1 Purpose

Each year, in living rooms and home-goods stores across the country the same battle unfolds: White or multi-colored lights? The battles are contentious with husbands and wives, neighbors and colleagues pitted against each other and searching for compromise.

In addition, this issue becomes bigger when major stores like Amazon, want to attract their customers during holidays. The choice of light colors may affect the customer's decision to visit or not to visit the store!

In order to solve this issue, and to have some fun, we decided to implement a system that controls the holiday lights' color by analyzing the social feeds and posts on twitter.

The purpose of this document is to elicit the software requirements of this project.

1.2 Document Conventions

Each requirement will be having an ID and it will be related to a specific group. These requirements have different priorities [2]. The requirements priorities will be mentioned as well.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read and used by developers, project managers, marketing staff, users, testers and documentation writers. We do advice the developers to skip some of the sections and go directly to the features and non-functional requirements part of this document. For the project managers and marketing staff, it is better for you to read the whole document, focusing on the user & product scope.

1.4 Product Scope

The goal of this product is to help the stores to gain more sales by attracting more customers during the holiday periods [1]. Providing an interaction between the clients and their store through Twitter will result in this store being a trending topic on twitter and everyone will be tweeting about it. The clients will tweet and mention their store in order to see their favorite holiday lights in action.

1.5 References

[1] Holiday Sales Statistics, satista.com
[2] IEEE RD conventions

2. Overall Description

2.1 Product Perspective

This product is a brand-new software in the market. It can be categorized as a Marketing/Sales product, more than a software engineering product. Once the product is developed and installed, no more software interaction and programming will be needed. Meanwhile, The maintenance side and the software life-cycle part will be continued. In addition, some store specific features can be added to the software when requested, like adding new functionalities. E.g. Controlling the products place within the store through a twitter pol.

2.2 Product Functions

The main product functions which the client and the store will benefit from are listed below.

- Clients suggesting holiday lights colors to the store through twitter.
- Store related tweets monitoring
- Store related tweets analysis (holiday lights color related)
- Automation of holiday lights color
- Listing the favorite holiday lights color for each store

2.3 User Classes and Characteristics

Totally, the system will have 3 user classes:

- **The Client Perspective:** The people or the store’s client who will be tweeting about their favorite holiday lights color.
- **The Store Perspective:** The people or the store’s management who will be monitoring the hardware part of this project and where the lights are having place.
- **The Non-Client Perspective:** The croud who are not interested in the holiday lights of stores. They can however for fun, check the favorite holiday light color of each store by visiting a dynamic web page.
- **The Developer Perspective:** The people or the person responsible about the software side development. Maybe the IT team in the store. These type of users are required in order to add any required new functionalities to the system.

2.4 Operating Environment

The software will be operating on Raspberry Pi 3, which will be connected to the holiday lights within the store. The Raspberry Pi 3 is running Linux as an operating system and the backend side is developed in Python 2.7. Some cables and breadboards will be needed to connect the Raspberry Pi 3 to the Holiday Lights. For any web development part, Python or JS frameworks will be used.

2.5 Design and Implementation Constraints

We believe that the following constraints will be available.

- **Hardware limitations:** We need the system to be using a Raspberry Pi 3 for financial concerns.
- **Timing requirements:** We need the system to be done within 3-4 months.
- **Security concerns:** We need to have a secure system, where the clients can not cause damage to our twitter account through this device.
- **Database Type:** The database should be MySQL as our IT team is experienced with it.
- **Programming Languages:** Only Java, JS, Angular, and Python can be used for this project. Our team can not handle any other programming language.
- **Design document:** The design document should be handed to the client, as they may integrate some new features to the system.

2.6 User Documentation

In order for you to get the needed support and discussions, you can visit our github page and our Trello Board. Please send as an email regarding your issue and then we can add you as a contributor to these two platforms. Please check the Appendix.

3. External Interface Requirements

3.1 User Interfaces

For our system, we will have three type of user interfaces mentioned below.

3.1.1 Client's User Interface

The only interface for the store's clients people will be their twitter account. No actual GUI is needed as they will be interacting with the system through twitter.

3.1.2 Store's User Interface

The store will have a Web application used only to inject his twitter account details (the store's account) so that the Raspberry Pi 3 can function properly.

3.1.3 Croud User Interface

A general user interface without a login page, will be available to the croud to check the favorite holiday lights of each store.

3.2 Hardware Interfaces

The only hardware interface of this project is the Raspberry Pi 3. It will act as an intermediary between the Python Program (software side) and the holiday Lights within the store. The data will be the users' tweets.

3.3 Software Interfaces

The software interfaces used for this project are mentioned below.

- The Linux operating system will be the host of all the software interfaces.
- An API in the Backend side will control the data flow in and out from the internet to the database.
- The database will be MySQL which is communicating directly with the Python program on the Raspberry Pi 3.
- The Website (for the croud) will be communicating with the database through the backend API.

3.4 Communications Interfaces

All the communications between the software interfaces will be communicated through the HTTPs and FTPs protocols. Please note that all of these connections should be encrypted for security reasons.

4. System Features

In this section, we will be discussing the functional requirements of this product.

4.1 Tweets Collection

4.1.1 Description and Priority

This is the most critical requirement. The tweets should be collected by the Python program at the very first hand. These tweets are the the ones addressed to the specified store. The program should be able to distinguish these tweets relatively. For example, a tweet for Amazon store, should not fall in the ebay store tweets.

4.1.2 Stimulus/Response Sequences

The twitter user mentioning the store in a tweet and recommending his favorite holiday lights color. For example: *@Amazon my favorite holiday color is White!*

4.1.3 Functional Requirements

REQ-4.1.3.1: The program will monitor the tweet mentions on the store's twitter page then collect them as a software object. It will be doing so for 23 hours each day, from 00:00 until 23:00.

REQ-4.1.3.2: The program should be able to distinguish between the tweets targeting the store in study, and the tweets targeting other stores.

REQ-4.1.3.3: The program should be able to distinguish between the tweets related to the Holiday lights color the ones who are not. Some domain should be established. For example: *@Amazon* ***my favorite holiday color*** *is White!*
@Amazon ***I love the*** *colorful* ***holiday lights****!*

4.2 Tweets Analysis

4.2.1 Description and Priority

This is a requirement with high priority. The tweets collected by the program should be analyzed to decide the winning holiday color for the store in study.

4.2.2 Stimulus/Response Sequences

The tweets collected in 4.1.

4.2.3 Functional Requirements

REQ-4.2.3.1: The program will analyze the collected tweets and specify whether it is recommending the white holiday lights or the colorful holiday lights. This will be done between 23:00 and 23:59 on each day.

REQ-4.2.3.2: The program will count the tweets recommending the white holiday lights and the ones recommending the recommending the colorful holiday lights.

REQ-4.2.3.3: The program will decide the winning color, which is one having the most tweets.

REQ-4.2.3.4: The program will store the result as an object.

4.3 Winner Color Activation

4.3.1 Description and Priority

This is a requirement with high priority. After the winner color is decided in 4.2, the program should change the holiday lights color at 00:00 of the next day.

4.3.2 Stimulus/Response Sequences

The decision made in 4.2. The Winner Color.

4.3.3 Functional Requirements

REQ-4.3.3.1: The program will change the holiday lights color to colorful, if the colorful vote won, or to white if the white vote won. This should be done at 00:00 of the next day.

4.4 Tweets database storage

4.4.1 Description and Priority

This is a requirement with low priority. All the analyzed results, and all the tweets should be stored to the database.

4.4.2 Stimulus/Response Sequences

The decision made in 4.1, 4.2 and 4.3

4.4.3 Functional Requirements

REQ-4.4.3.1: The collected tweets and winner color for each store should be stored within the database. The tweets of the different stores should be separated from each other.

4.5 Providing a simple web interface for the croud

4.5.1 Description and Priority

This is a requirement with low priority. After the winner color is decided by the end of each day, a webpage should show the winner color related to each store in a tabular format. For example: Amazon --- colorful
eBay --- white

4.5.2 Stimulus/Response Sequences

The decision made in 4.3

4.5.3 Functional Requirements

REQ-4.5.3.1: The program will provide an web interface for the croud in order to check the winner color for each store, on each day. It should be updated regularly on each day.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall be both resource-efficient and scalable. That is, as the number of users increases, the system, by adding new resources, should work as if there is only one user. This scalability should be satisfied in a distributed manner.

- the system shall have responsive user interfaces (for the croud) so that it can be more accessible. It shall support modern web browsers such as Firefox or Chrome and their corresponding mobile versions.

- The system shall use and contain only open source technologies, libraries, and tools.

- The system shall expose an HTTP RESTful API which supports every user interactions which can also be done via the user interfaces of the system. This API shall have complete documentation.

5.2 Safety Requirements

The system should not cause any damage to the store, or to the holiday’s lights connected to it.

The system should be safe to use, and reporting any abnormal temperature changes within the lights or the raspberry pi 3.

5.3 Security Requirements

- The raspberry Pi 3 and the Python program should as secure as possible. Usage of encryption in all the data processing parts will be required.

- The twitter account connected to the raspberry pi 3 should not get any damage from the users' tweets. It should be only collecting their tweets and ignoring any unsuitable ones.

Appendix A: User Resources

Github page: https://github.com/yasserkabbout/tweeLights
Trello page: https://trello.com/b/9anSOIJD/tweelights