McMaster University

SFWRENG 4G06

Tip of My Shoe - SRS

Member Name	Mac ID	Student Number
Chris Dibussolo	dibussoc	400070368
Andrew Lucentini	lucenta	001430150
Owen McNeil	mcneilo	400065750
Daniel Scime	scimed1	400069926
Ashley Williams	willia18	400081787
Lucas Zacharewicz	zacharel	400054446

October 27, 2019

Contents

1 Overview **2**

1.1 Purpose . . . 2

1.2 Scope . . . 2

1.3 Stakeholders . . . 3

1.3.1 The Client . . . 3

1.3.2 The Customers . . . 3

1.4 Normal Operation/Use Case . . . 3

1.5 Context Diagram . . . 3

1.6 Monitored Controlled Variables . . . 3

1.6.1 Monitored Variables . . . 3

1.6.2 Controlled Variables . . . 3

1.6.3 Constants . . . 3

1.6.4 Relevant Facts . . . 3

1.6.5 Relevant Assumptions . . . 3

2 Functional Requirements **3**

2.1 Functional Requirements and Behaviour Overview . . . 3

2.1.1 Overview of Product Behaviour . . . 3

2.1.2 Functional Decomposition . . . 4

2.1.3 Required Functions . . . 4

3 Non-functional Requirements **5**

3.1 Usability and Humanity Requirements . . . 5

3.1.1 Ease of Use Requirements . . . 5

3.1.2 Personalising and Internationalization Requirements . . . 6

3.1.3 Learning Requirements . . . 6

3.1.4 Understandability and Politeness Requirements . . . 6

3.1.5 Accessibility Requirements . . . 7

3.2 Performance Requirements . . . 7
3.2.1 Speed and Latency Requirements . . . 7
3.2.2 Safety-Critical Requirements . . . 7
3.2.3 Precision or Accuracy Requirements . . . 8
3.2.4 Reliability and Availability Requirements . . . 8
3.2.5 Robustness or Fault-Tolerance Requirements . . . 8
3.2.6 Capacity Requirements . . . 9
3.3 Security and Legal Requirements . . . 9
3.3.1 Integrity Requirements . . . 9
3.3.2 Privacy Requirements . . . 10

4 Undesired Scenario Handling **10**
4.1 The System Cannot Extract Information from the Image . . . 10
4.2 The System Cannot Find Close Matches . . . 10
4.3 Software Failure (General) . . . 10

5 Possible Changes and Noteable Risks **11**
5.1 Requirements That Are Likely to Change . . . 11
5.2 Requirements That Are Not Likely to Change . . . 11
5.3 Risks . . . 11
5.3.1 Excessive Schedule Pressure . . . 11
5.3.2 Management Malpractice . . . 11
5.3.3 Low Quality . . . 11
5.4 Costs . . . 11

6 References **12**

1 Overview

1.1 Purpose

The purpose of this document is to present a functional and non-functional description for the software application titled **Tip of My Shoe**. The document shall elaborate on the overall description, functional requirements, and non-functional requirements of the software. This document is intended for developers, stakeholders, and users of the application

1.2 Scope

Tip of My Shoe will allow users to provide a picture of a shoe, in order to receive an identical/similar list of shoes provided via web-store link as to where the shoe can be purchased. The shoes will be sorted by highest matching. The project is intended (but not limited) to be administered as a smart-phone application. The project will incorporate numerous front-end and back-end technologies in-order to create a fully functioning user interface. Specific functionality is provided in the Functional Requirements section of this document.

1.3 Stakeholders

1.3.1 The Client

1.3.2 The Customers

1.4 Normal Operation/Use Case

1.5 Context Diagram

1.6 Monitored Controlled Variables

1.6.1 Monitored Variables

1.6.2 Controlled Variables

1.6.3 Constants

1.6.4 Relevant Facts

-

1.6.5 Relevant Assumptions

The target user of this product is assumed to be:

-

2 Functional Requirements

2.1 Functional Requirements and Behaviour Overview

2.1.1 Overview of Product Behaviour

Tip Of My Shoe will provide users a list of shoes (via web-store links) that match a shoe that the user desires to purchase. The user will submit the design via a picture of a shoe. The software should match the patterns and overall design of the shoe to provide the user with results of retail products that would provide a similar look.

2.1.2 Functional Decomposition

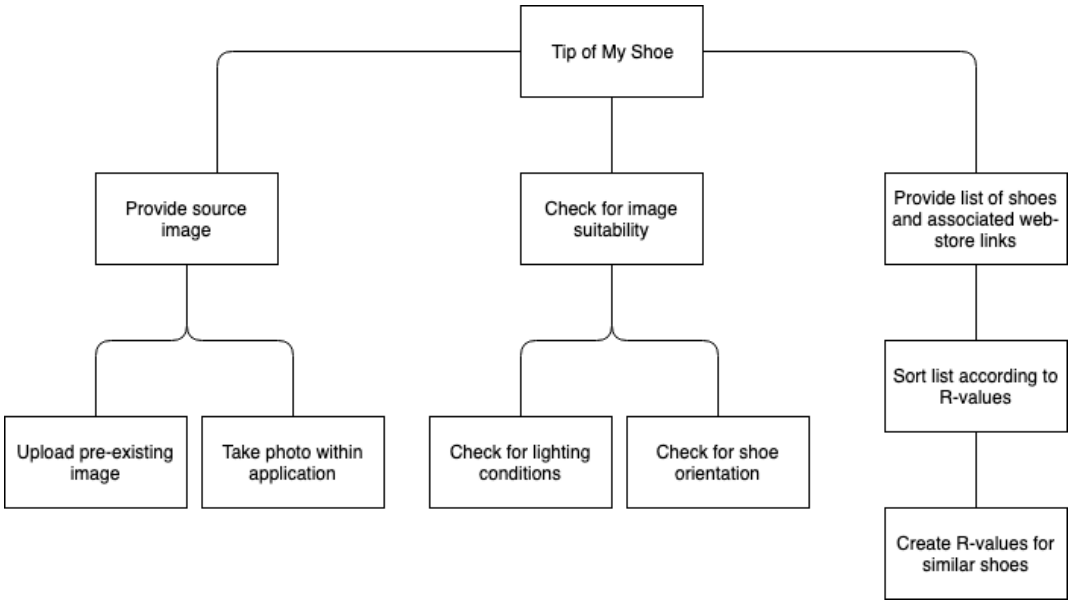

Figure 1: Functional Decomposition Diagram

2.1.3 Required Functions

- **FR1:** The system will allow the user to upload an image or take a picture within the application.

  Rationale: The user may already have an image of the shoe they are interested in, or may want to take their own photo without storing it on their device.

- **FR2:** The system will only accept images of a specific form (e.g. suitable lighting conditions, shoe orientation) and will notify the user when an image is not usable.

  Rationale: A clear photo of the shoe must be provided in order to identify it. Additionally, images provided by retailers are typically taken from specific views/orientations. Since identification will be based on these images, the source image should be in a similar form.

- **FR3:** The system shall provide a list of shoes to the user that resemble some source image of a shoe.

  Rationale: This is the main goal of the project, however, some liberties may be taken. For example, if it proves too difficult to provide multiple matches that resemble the shoe in the

image to create a similar "look", we may opt to make the goal to find a retail seller of the exact shoe in the image should it be available. The subjective nature of matching the "look" of a shoe may prove much more difficult than definitively matching a set of parameters.

- **FR4:** Provided shoes should have an associated "related value" (R-value). This R-value measures how similar the resulting shoe is to the searched shoe.

  Rationale: This value will put a measure to the subjective concept of "how much does this shoe look like what I want?".

- **FR5:**

  Rationale:

3 Non-functional Requirements

3.1 Usability and Humanity Requirements

This section is concerned with requirements that make the product usable and acceptable to the users. This is an important factor pertaining to our project.

3.1.1 Ease of Use Requirements

- **Requirement:** The software should be usable by all potential customers within reason. The target age is 8+ years old.

  Rationale: Maximizing the domain of users will allow for large data collection, hence more accurate results.

- **Requirements:** The user should be able to easily operate the camera with minimal instructions.

  Rationale: The camera is the main means of searching for shoes based on an image taken by the user (a picture not taken from the internet or some secondary source). The camera

should have on-screen instructions and automated aid for the user so minimal issues are experienced.

- **Requirements:** The interface should provide a simplistic sense of navigation through the program.

  Rationale: To promote repeated use, the user should not experience any issues in trying to figure out how to do what they want to do.

3.1.2 Personalising and Internationalization Requirements

- **Requirements:** Users should eventually be usable to people of multiple nationalities, however, will predominantly focus on a western release, focusing on USA and Canada accessibility for anglophone users.

  Rationale: Given that the main challenge is making the software work, internationalization should be a task left for post. Initially, resources should be allocated to focusing on development for strictly English users in North America.

3.1.3 Learning Requirements

- **Requirements:** The user should require minimal learning to understand how to use the software. On-screen instructions and intuitive interfacing should provide a seamless experience for the user.

  Rationale: Inherently, the use of the product is fairly simple, all the hard work is done by the software. All the user has to do is supply an image for their search. The process of supplying said image should therefore be made as easy as possible for the user so they do not have to learn anything complicated about how the software works.

3.1.4 Understandability and Politeness Requirements

- **Requirements:** The overall usage and purpose of the software should be made easily apparent to the user. The user should not have to wonder what they are using the software for.

Rationale: Given the niche nature of the software, it should be made very apparent to the user what exactly they are using it for.

3.1.5 Accessibility Requirements

- **Requirements:** The software should be accessible by anyone with a smartphone or computer on a number of operating systems.

  Rationale: Primarily, the software should be available and provide solid performance on Windows 7-10 and MacOSX for PC users as well as Android and IOS for mobile users, given that these are the most likely ways for our target audience to be using the software.

3.2 Performance Requirements

This section is concerned with requirements regarding the performance of the system.

3.2.1 Speed and Latency Requirements

- **Requirement:** The software's search process should take no longer than 5 seconds to complete.

  Rationale: A search process longer than 10 seconds will make the user think the software is unreliable and they will be unlikely to use again. Similarly, if the user is waiting for 10 seconds for every image they search, they will be discouraged from using the software extensively.

3.2.2 Safety-Critical Requirements

- **Requirement:** The software must not put excessive strain on the CPU so as to overheat the hardware.

  Rationale: We want to minize the risk of a hardware faiure. This can be accomplished by minimizing the strain on the CPU.

3.2.3 Precision or Accuracy Requirements

- **Requirement:** The system must return shoes with R-values greater than 0.8.

  Rationale: The system must provide meaningful results.

- **Requirement:** A list of similar shoes shall be provided to users if the system is unable to find an identical match.

  Rationale: In the case that a shoe cannot be found, we must provide the users with meaningful results.

- **Requirement:** The similarity between the desired (user searched) and resulting shoe/shoes must be measurable by a defined scale through a combination of factors.

  Rationale: By measuring the "similarity" between desired and resulting shoe/shoes, we are able to identify the accuracy and efficiency of the system.

3.2.4 Reliability and Availability Requirements

- **Requirement:** The software should maintain 99% up-time.

  Rationale: This percentage means that for every 100 days only 24 hours of downtime should be expected. This time is left for maintenance and future patches of the software. Excessive down-time will make users unlikely to ever use the product given the niche nature of the product. Most people will assume it is too convenient to be true, and experiencing downtime will make them give up any expectation.

3.2.5 Robustness or Fault-Tolerance Requirements

- **Requirement:** In the event of an error, particularly due to user input, the software should be able to handle the error without crashing and administer appropriate feedback to the user to ensure appropriate input is received.

  Rationale: A robust software should be able to handle undesired user input without a catastrophic system crash. To avoid recurring errors the system should provide the user with

appropriate signifiers and feedback to ensure minimal errors occur. This also reinforces the desired seamless usability of the software.

- **Requirement:** In the event of a catastrophic failure (unhandled failure), the application should identify the failure upon re-opening.

  Rationale: A user who experience a program crash with no sense of aid from the system or instructions to proceed towards recovery is likely to not use the product again.

3.2.6 Capacity Requirements

- **Requirement:** The system shall handle the processing of 10,000 simultaneous users.

  Rationale: Based on expected user growth, allowing the system to handle 10,000 simultaneous users will ensure that the system does not need functional modification for the years to come.

- **Requirement:** Server space will be required to support user accounts and store user data for the sake of machine learning and increasing the data set for the machine learning process.

  Rationale: Machine learning requires extensive data sets and along with web scraping it is likely that it will be necessary to store user data to make these data sets robust, making the search as accurate as possible.

3.3 Security and Legal Requirements

3.3.1 Integrity Requirements

- **Requirement:** The system must ensure that all information will be stored on servers via encryption.

  Rationale: Should the software support user account data be secure to avoid loss of system credibility and endangering the user, especially if important information such as location is saved to the user account. Should the system be successful and a subscription fee be hypothetically in the future, the encryption of a user's financial information would also be critical.

3.3.2 Privacy Requirements

- **Requirement:** The system must ensure that no privacy laws are broken.

  Rationale: Obvious reasons. We don't want no trouble.

4 Undesired Scenario Handling

The following sub sections pertain to scenarios in which the system may not behave correctly, and how corrective action will be taken in these situations.

4.1 The System Cannot Extract Information from the Image

There may be situations in which the picture provided to the system is inadequate for the system to extract information. The system will be able to identify this situation, and request that users provide a "better" image. The system will also provide the reason for error (i.e "This image is too dark").

4.2 The System Cannot Find Close Matches

If the system cannot find any close matches, it must the system must notify the user of this case. In addition, the system must provide shoes that have a lower R-Value than 0.8.

4.3 Software Failure (General)

If the system fails due to an unexpected software failure, the system must notify the user that the application has crashed once the user re-opens the application.

5 Possible Changes and Noteable Risks

5.1 Requirements That Are Likely to Change

5.2 Requirements That Are Not Likely to Change

5.3 Risks

5.3.1 Excessive Schedule Pressure

Due to the nature of the project and the teams inexperience with machine learning software, maintaining a manageable schedule may prove daunting, due to the unpredictability of any said tasks time consumption.

5.3.2 Management Malpractice

The team must ensure that they stay organized on their management of time and resources while working on the project as falling behind make them unable to deliver.

5.3.3 Low Quality

Excessive focus on fancy features beyond the scope of the project will lead to the team being overwhelmed and producing low quality software. The team should avoid this risk by focusing on consistency and reliability, making sure the simple features work properly, instead of producing unneeded features that create more problems.

5.4 Costs

6 References