**Software Requirements**

**Specification**

**for**

**Smart Eye Drops**

**Version 2.1 approved**

**Prepared by**

**Prepared by Aaron Alvarez, Marco Aguilar, Ana Cortes, Jonathan Nunez,**

**Feng You, Ivan Yu**

**Vodafone**

**March 18, 2020**

Table of Contents

[Revision History 4](#_Toc39847739)

[1. Introduction 5](#_Toc39847740)

[1.1 Purpose 5](#_Toc39847741)

[1.2 Intended Audience and Reading Suggestions 5](#_Toc39847742)

[1.3 Product Scope 6](#_Toc39847743)

[1.4 Definitions, Acronyms, and Abbreviations 6](#_Toc39847744)

[1.5 References 6](#_Toc39847745)

[2. Overall Description 8](#_Toc39847746)

[2.1 Product Perspective 8](#_Toc39847747)

[2.2 Product Functions 8](#_Toc39847748)

[2.3 User Classes and Characteristics 8](#_Toc39847749)

[2.4 Operating Environment 9](#_Toc39847750)

[2.5 Design and Implementation Constraints 9](#_Toc39847751)

[2.6 User Documentation 9](#_Toc39847752)

[2.7 Assumptions and Dependencies 9](#_Toc39847753)

[2.8 Apportioning of Requirements 9](#_Toc39847754)

[3. External Interface Requirements 10](#_Toc39847755)

[3.1 User Interfaces 10](#_Toc39847756)

[3.2 Hardware Interfaces 10](#_Toc39847757)

[3.3 Software Interfaces 10](#_Toc39847758)

[3.4 Communications Interfaces 11](#_Toc39847759)

[4. Requirements Specification 12](#_Toc39847760)

[4.1 Functional Requirements 12](#_Toc39847761)

[4.2 External Interface Requirements 12](#_Toc39847762)

[4.3 Logical Database Requirements 13](#_Toc39847763)

[4.4 Design Constraints 13](#_Toc39847764)

[5. Other Nonfunctional Requirements 14](#_Toc39847765)

[5.1 Performance Requirements 14](#_Toc39847766)

[5.2 Safety Requirements 14](#_Toc39847767)

[5.3 Security Requirements 14](#_Toc39847768)

[5.4 Software Quality Attributes 14](#_Toc39847769)

[5.5 Business Rules 14](#_Toc39847770)

[6. Other Requirements 15](#_Toc39847771)

[Appendix A: Glossary 16](#_Toc39847772)

[Appendix B: Analysis Model 17](#_Toc39847773)

[Appendix A: To Be Determined List 18](#_Toc39847774)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Ivan Yu | 11/28/2019 | Initial creation | 1.0 |
| Marco Aguilar | 02/26/2020 | First revision | 2.0 |
| Marco Aguilar | 03/18/2020 | Final revision | 2.1 |

## 1. Introduction

This SRS (Software Requirements Specification) is a document used to describe the purpose, specifications, functional and non-functional requirements of Smart Eye Drops. This document will also go over the hardware implementation such as our sensors. Smart Eye Drops is a system that will assist physicians in increasing eye drop adherence. The project was inspired by the statistics of traditional medication versus eye medication. The percentage of people who remain obedient to traditional medication is significantly higher than eye care. We attempt to solve the issue of deficient eye drop adherence with Smart Eye Drops.

### 1.1 Purpose

The purpose of Smart Eye Drops is to increase the adherence rate of eye medication. This is Smart Eye

Drops version 2.0. The purpose of this SRS is to serve as the main source of requirements for **Smart Eye**

**Drops** programmers who will write the SDD and then implement the code that will become **Smart Eye Drops** software. This is the only authorized source of **Smart Eye Drops** software requirements. The document’s first version is the one being reviewed and approved. The second version will be the one including all the initial inputs from the reviewers and the one used to design implement (code) the software. The third and final version of **Smart Eye Drops** SRS will include all the modifications approved after the SDD, if any.

The requirements will be listed using tables with entries for each module initially understood to be part of **Smart Eye Drops**. These tables may be modified accordingly after the SDD has been generated and approved. Lastly, there may be requirements that will be better understood at the implementation stage and these will be the last modifications made to the requirements table before delivery.

### 1.2 Intended Audience and Reading Suggestions

This document is intended for the developers, project manager, users, testers, and documentation writers. Those who wish to update or review the documentation (e.g. documentation writers) should go to the table of contents to get a better understanding of the overall structure of the SRS. Future developers and product managers of the Smart Eye Drops system should refer to section 2. This section provides valuable information about the major functions of the system, software environment, and constraints along with the limitations of the Smart Eye Drops software.

Smart Eye Drops is designed to assist physicians with a patient’s compliance to eye medication. Smart Eye Drops sensors will collect data and send it to the physician. Doctors can then adjust the patient’s regime and give tips to patients about how they can improve adherence rate. Additionally, the device can help improve research on eye medication. Smart Eye Drops is primarily focused on serving doctors and patients on improving adherence percentage.

### 1.3 Product Scope

Smart Eye Drops will monitor each individual user while they use their eye medication. The data collected will then be sent over to the cloud service to be preprocessed and classified with the help of machine learning algorithms. The classified data will be stored in the database and then sent to physicians whenever they open the Smart Eye Drops application. The software will also give tips to users on how to improve adherence. Smart Eye Drops is intended to be used to monitor the user activity and compliance to eye medication. A benefit of this product will be that users will improve their adherence percentage. An increase in adherence will reduce the chances of the patient developing eye problems.

### 1.4 Definitions, Acronyms, and Abbreviations

The definitions, acronyms, and abbreviations used throughout this document can be found in Appendix A, the Glossary portion of the document, located on page 16.

### 1.5 References

This SRS template.

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

Last, F. M. (Year, Month Date Published). *Article title*. Retrieved from URL

Vision Eye Institute (2018, August 08). *Eye drop misuse is threatening the sight of 150,000 Australians with glaucoma*. https://visioneyeinstitute.com.au/eyematters/eyedrop-misuse-glaucoma/

L, Jeff. (2018, May 16). *15 Frightening Stats on Medication Adherence (Plus Infographic)*.

https://www.pillsy.com/articles/medication-adherence-stats

Sleath, B., Blalock, S., Corvert, D., Stone, J. L., Skinner, A. C., Muir, K., Robin, A. L. (2011, December 01). *The relationship between glaucoma medication adherence, eye drop technique, and visual field defect severity*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3223548/

## 2. Overall Description

### 2.1 Product Perspective

There are other projects that are like Smart Eye Drops. However, most of those products never reached the market. Smart Eye Drops is the first one to do it wirelessly and detect squeezing of the eye drop bottle whilst occupying a small footprint. Other similar products also do not have a target audience. We have an idea and an image behind our product. We intend to send compliance data to physicians. Along with sending data to physicians, we also intend to send tips to help increase adherence percentage amongst patients.

### 2.2 Product Functions

We will be using the MMC sensor alongside a FSR in our device to collect data from the patients. The data will be transmitted via Bluetooth to a hub or a mobile device. We will then send that data to AWS to process with machine learning algorithms. Once that is complete, we will then return data to the physicians and patients.

* Smart Eye Drops will collect compliance data from patients by logging when they are using the eye drops.
* Smart Eye Drops will collect adherence data from patient by collecting data on how the patient is applying the eye drops.
* Smart Eye Drops will send data to AWS to process through machine learning algorithms.
* Smart Eye Drops will return adherence and compliance data to physicians.
* Smart Eye Drops will return tips on how to improve adherence to the patients.

### 2.3 User Classes and Characteristics

The user class that belongs to Smart Eye Drops are patients that require eye medication and doctors. This product is mainly intended for, but not limited to, doctors that are trying to monitor the patient’s eye care. A primary example for our user class is glaucoma patients. Statistics indicated that 44% of patients stop compliance to their eye care regime within the first six month. This is not the sole purpose of Smart Eye Drops. The product can also be used to assist in the research of eye medication. Smart Eye Drops can even be used generally by users who want to improve adherence of over-the-counter eye drops (such as artificial tears).

### 2.4 Operating Environment

The operating system for Smart Eye Drops is Android (version 22 or higher). We will be using AWS services, such as Lambda, EC2, and DynamoDB, to store all our data. Android will be our primary platform for physicians and patients alike. Physicians can use the app to look at patients’ compliance data and patients can use the android app for adherence tips.

### 2.5 Design and Implementation Constraints

* Smart Eye Drops require AWS cloud storage to store all collected data.
* Smart Eye Drops require to be close to a Bluetooth device (hub/mobile device) to send data to the cloud.
* Internet connection is required to send data to the cloud.
* Smart Eye Drops require a device to look at compliance and adherence data.
* Smart Eye Drops require MBientLab sensors and pressure sensors to collected data from patients.

### 2.6 User Documentation

Smart Eye Drops SRS Document.

### 2.7 Assumptions and Dependencies

Smart Eye Drops require a machine learning algorithm that is accurate enough to provide accurate predictions in terms of compliance. This is also assuming that all data collected is accurate. Smart Eye Drops also require internet connection, and an android device. Without these, the project will not work. One of the dependencies that our system has is the MBientLab sensors. These sensors provide the functionality needed to collect live data from the eye drop container. A change in hardware would require the SRS to be reviewed and revised.

### 2.8 Apportioning of Requirements

1. Properly connecting the sensors to the eye drop bottle without the use of a breadboard.
2. Fully functional machine learning capabilities.

## 3. External Interface Requirements

### 3.1 User Interfaces

|  |  |
| --- | --- |
| User Interface | Description for User Interface |
| 3.1.1 Visualize | Used to request for a visualization of compliance and adherence data. |
| 3.1.2 Physician Request Data | Physician request for patient’s data. |
| 3.1.3 Patient Adherence Chart | Displays a patient’s adherence for a specific month. |

### 3.2 Hardware Interfaces

|  |  |
| --- | --- |
| Hardware Interface | Description for Hardware Interface |
| 3.2.1 Sensors | Used to collect data from the patient. |
| 3.2.2 Android Device | Used to send and receive data from the cloud. |

### 3.3 Software Interfaces

|  |  |
| --- | --- |
| Software Interface | Description for Software Interface |
| 3.3.1 Java | Used to program application in Android Studio, send and collect data from our hardware. |
| 3.3.2 Android Studio | Used to create Android Application for Patients and Physicians |
| 3.3.3 AWS | Used to store data into the cloud and eventually process data through machine learning algorithms. |
| 3.3.4 Jupyter Notebook | IDE for Python |
| 3.3.5 Python | Used for processing machine learning algorithms. |

### 3.4 Communications Interfaces

|  |  |
| --- | --- |
| Communication Interface | Description for Software Interface |
| 3.4.1 API Gateway | Used for creating a REST API. Enables HTTP communication between AWS services. |
| 3.4.2 Lambda | Used to create serverless application. Allows us to make changes to our database. |
| 3.4.3 EC2 | Provides secure, resizable compute capacity. Used for processing raw data before sending out to other AWS services. |
| 3.4.4 SageMaker | Used for hosting all our machine learning models and endpoints. Allows us to make a prediction by sending it data. |
| 3.4.5 DynamoDB | Cloud hosted noSQL database. Allows key-value data structures. Stores all our data for future applications. |

## 4. Requirements Specification

### 4.1 Functional Requirements

4.1.1 The system shall receive sensor input from the user

4.1.2 The system shall remove noise from the sensor input

4.1.3 The system shall transmit live data to the cloud service

4.1.4 The system shall classify and store data

4.1.5 The system shall transfer data to the physician’s mobile device

4.1.6 The system shall display the adherence data point to the physician via a chart

4.1.7 The system shall deal with any errors that occur

### 4.2 External Interface Requirements

|  |  |
| --- | --- |
| H-1 | Collect accelerometer data. |
| H-2 | Collect gyroscope data. |
| H-3 | Collect pressure monitor data. |
| H-4 | Collect ultrasonic monitor data. |
| H-5 | Send/store all data. |

### 4.3 Logical Database Requirements

|  |  |
| --- | --- |
| LD-1 | Timestamps are stored and later used to determine compliance. |
| LD-2 | Accelerometer data are stored and later used to determine compliance and adherence. |
| LD-3 | Gyroscope data are stored and later used to determine compliance and adherence. |
| LD-4 | Ultrasonic monitor data are stored for later determining compliance and adherence. |
| LD-5 | Patient records are stored to keep track of the Physician’s clients. |

### 4.4 Design Constraints

TBD.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

The system should be able to handle all the live data that is collected from the eye drop container. In addition, the storage size should be large enough to hold all the data necessary to properly train a machine learning model. It is crucial that 95% of the data is classified in a timely manner so that the Smart Eye Drop system may transmit the information to the mobile application. Any delays lasting more than 24 hours will cause a bad user experience.

### 5.2 Safety Requirements

Not Applicable

### 5.3 Security Requirements

Minimal security is required. The only data collected are data and time, force applied, and values from the 3-axis gyroscope. No personal information shall be collected so the only security feature that will be implemented will be to ensure the correct data values are collected and not tampered with.

### 5.4 Software Quality Attributes

It is crucial that the Smart Eye Drop system can reduce the level of noise that is gathered from the sensor input. Noise will hinder the machine learning model’s ability to correctly classify the data that is collected from the user’s interaction with the eye drop container. This will inevitably lead to the physician receiving the wrong information about their patient.

### 5.5 Business Rules

Patients who use the software will not be able to make alterations to their regimen, which is assigned by their physician who can view the patience adherence ratings and make necessary changes to improve results.

## 6. Other Requirements

There are no other requirements at his time. This may change in the future as the implementation of the application progresses.

# Appendix A: Glossary

**AWS**: Amazon Web Services provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. https://aws.amazon.com/

**DynamoDB**: Cloud database that stores data for the application.

**SageMaker**: Hosting cloud service that allows the use of machine learning models and endpoints.

**Lambda**: Hosting cloud service that allows the use of machine learning models and endpoints.

**FSR**: Force Sensitive Resistors are sensors that allow you to detect physical pressure, squeezing and weight. https://learn.adafruit.com/force-sensitive-resistor-fsr/overview

**IDE:** Integrated Development Environment, a software application that provides comprehensive facilities to computer programmers for software development.

**MMC**: MetaMotion C is a wearable device that offers real-time and continuous monitoring of motion and environmental sensor data. https://mbientlab.com/metamotionc/

**MMR**: MetaMotion R is a wearable device that offers real-time and continuous monitoring of motion and environmental sensor data includes a rechargeable battery. <https://mbientlab.com/metamotionr/>

# Appendix B: Analysis Model

**SED**: Smart Eye Drops, eye drop bottle containing an **FSR** and **MMR** integrated with software to enhance adherence index calculation.

**API**: Application programming interface, tools that helps us gather data from the **MMR**.

# Appendix A: To Be Determined List

TBD.