# *Requirements Analysis Document*

# *Project:* Visual Field Exploratory Analysis

Semester Two 2019

University of Western Australia

---

**Revision History:**

Version R0.1 12/08/2019 Alex Rohl created.

**Preface:**

> This document addresses the requirements of the Visual Field Exploratory Analysis project. The intended audience for this document are the developers, clients and relevant stakeholders of the project.

**Target Audience:**

Client, Developers, Stakeholders

**Developers:**

Sam Dixon
Fraser Loneragan
Julian Kristandi
Jordan Harun
Justin Andronicos
Alex Rohl

**Clients:**

Dr Wei Liu, *Senior Lecturer EMS*
Dr Nigel Morlot, *Opthamologist*
Dr Jonathan Ng, *Opthamologist*
Siobhan Manners, *Senior Data Linkage Officer*

**Stakeholders:**

Aonghus Grealnish**,** *Data Science Masters Student*
May Hu, *Data Science Masters Student*

**MILESTONES**

- 21/08/2019 Sprint 1 due
- 20/09/2019 Sprint 2 due
- 25/10/2019 Sprint 3 due

# 1.0 General Goals/Scope

This project consists of three sprints, each over a four-week time period. Our goals for the first sprint are to acquire the desired level of knowledge of the domain and build a reliable, secure and organised database backend that can be interacted with via a front end web app. In order to acquire the desired level of knowledge, we intend to read dissertations, online articles and web guides in order to become familiar with relevant terms and tools such as DICOM, Flask, Visual Field, Humphrey Machines, Cloud Storage etc. Within the scope of our first sprint, we intend to define the relevant fields required for our database and host this in a structured manner on a reliable and secure cloud server. Furthermore, we intend to interface with the database through a web app that will initially allow us to pull data with respect to Patient_Name, Age and other interesting attributes.

Our goals for the second sprint are to create relevant visualisations that reflect the queries desired. Initially, this will require scripts to organise the compressed pixel data into a recognisable image displayable on the web app. Consequently, we intend to supply this organised image format to masters students for their classification algorithms using machine learning techniques. We intend to also provide a platform on which these classification algorithms can be run and consequent results be visualised. Such visualisations may reflect relationships between attributes, changes to variables over time or clustering among patient data.

Our goals for the final sprint will be to maintain the framework of the system such that new datasets can be securely uploaded to the database and stored in a universal format.

# 2.0 Current System

From the Humphrey Visual Field Machine, the doctors typically have an option for transferring a single selected test or multiple selected tests, or even all the tests. When transferring all the tests, it may take hours or even days, based on the number of tests on the machine. Once the tests are transferred onto a computer, they can be viewed in pdf format.

In general, the data is only being used to query a single patient and the results are only used to consult the relevant patient. However, since there are over 250,000 visual field analysis files that are currently not in any database system, there is room for exploration.

# 3.0 Proposed System

In order to derive the insights that potentially exist within looking at these files as an entire entity, we intend to create a database system that organises this data and a frontend web application as a platform to allow for efficient query capabilities. In particular, we will host this data on a SQL server as this is the format in which the data has been provided. Underlying the query process we intend to firstly run cleaning scripts using the python ‘pandas’ package to transform the image data into a more recognisable format and consequently adopt relevant classifiers that will sort the image data into the multitude of vision-related disorders.

## 3.1 Overview

In this section, we give a top-level description of each subsystem. In particular, the functional requirements which relate to objectively measurable tasks within the project and nonfunctional requirements which relate to elements of the project that are to be subjectively interpreted by the user.

## 3.2 Functional Requirements

- Secure Login
- Query the tabular database hosted on AWS cloud database service
- Implement tools to visualise the data
- Simple user interface

### 3.2.1 Possible Functional Requirements if time permits

- Create, Update and Delete files for future data.

## 3.3 Nonfunctional Requirements

Nonfunctional requirements which relate to elements of the project that are to be subjectively interpreted by the user.

#### 3.3.1 User Interface and Human Factors

We anticipate that our system will be used by a range of doctors, admins and researchers. Hence, it is crucial that our system is easy to use on both laptop and mobile devices. In particular, it is essential that the output of the system reflects what the user wants and has the intention of defining in their query. To improve the user experience, we intend to include interactive visualisations e.g. hovering over a graph to show a specific data point. For ease of use, we will include a user guide so that users can efficiently find accurate answers to their queries. For researchers' purposes, we will include a diagrammatic overview of the system and its server-database backend, for ensuring comprehension of the rigorous methods used.

Regarding security, given the sensitivity of the data, we need to ensure adequate data protection. Hence, we intend to incorporate admin credentials into the system.

#### 3.3.3 Hardware Consideration

Our website will be assessed through web browsers, therefore common platforms expected would include mobile phones, computers, and laptops. We will have to ensure that our webpage can run smoothly across various common web browsers.

#### 3.3.4 Performance Characteristics

We will be working with approximately 250,000 DICOM files, at 7 KB each, which totals to 1.75 GB of data. We have agreed with the clients that the program accuracy will be prioritised, rather than its performance or speed.

#### 3.3.5 Error Handling and Extreme Conditions

Error Scenarios / Handling:

- Multiple consecutive incorrect login attempts - timeout for login.
- Invalid query statements - notify users that format is incorrect or field does not exist.

Note - this list will increase as we finalise our website and potential for errors becomes more apparent.

### 3.3.6 System Interfacing and ETL Process

Input Methods and Requirements:

- Upload DICOM files.
- Ensure DICOM files are filled in with complete data.
- (potential) Methods to upload other related data (driving records etc.)

ETL Process:

We intend to store the data on a SQL Server. As such, we need appropriate methods to convert the data files into a database comprising of only the important fields. We intend to use a 'Star Schema' to represent the data relations and as such intend to include the following tables (refer to database design folder for a pictorial view):

- **Fact Table**
  - Will contain a key referring to the following tables, as well as the following measures:
    - Date
    - RunTime
    - (potential) Mean Deviation
    - (potential) Pattern Deviation
- Test Table
  - Name of Test
  - Type of Test
  - Area tested? (for future implementations of a broader range of health tests)
- Defects Table
  - We found there to be a large number of defect information
- Humphrey Image Table
  - Will need to manipulate the original 'StaticPointDataField' into an entire table consisting of columns corresponding to the coordinates of the image array.
- Reliability Table
  -
- EyeTable
  - EyeTested
  - Visual Acuity

### 3.3.7 Quality Issues

Quality Standards:

- Website needs to be up for 24 hours, 7 days a week.
- Ensure the accuracy of data displayed.
- Deploy appropriate error handling implementations.

### 3.3.8 System Modifications

Future Usage:

- Visualisation from DICOM Data (Based on final tabular format).
- Usage for all types of DICOM Files (not only for Visual Field purposes).

### 3.3.9 Security Issues

As we are working with highly sensitive data, we need to ensure sufficient security measures to protect the patient's confidentiality. We need to make sure that the data cannot be extracted or modified.

#### 3.3.10 Resource Issues

We are planning to use Amazon Web Services (AWS) as our main cloud server, so any further issues will be resolved with the third party.

## 3.4 Constraints

The programming languages that we are using would be Python, SQL, HTML, Javascript. We would use Flask as our main framework. Our client suggests us to use Jupyter Notebook to write the python aspects of our program. Some of the common web browsers we would be testing on include: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Internet Explorer.

## 3.5 System Model

The infrastructure of the system will include using AWS for our cloud storage, SQL for our main database (SQLAlchemy for our ORM), Python (Flask) for our back-end, BootstrapJS and HTML for our front-end.

#### 3.5.1 User Scenarios

See Stories List for details.

#### 3.5.2 User Interface - Navigational Paths and Screen Mockups

Main Pages:

- Login
- Admin Management (Request Admin Status to relevant authorities)
- User Page
- Uploading and Database Storage*
- Querying the Database (Search based on the available fields from the drop-down menu)
- User Manual (How to use the system)
- About Us

*We might not include this in the end; if time permits.