|  |
| --- |
| HDSSD |
| Requirements Specification (RS) |
| Migraine Diary |

|  |
| --- |
| Daniel Morrissey x21118701  6/21/2022 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
|  |  |
|  |  |

**Table of Contents**

[Requirements Specification (RS) 1](#_Toc106700701)

[Document Control 1](#_Toc106700702)

[Revision History 1](#_Toc106700703)

[Distribution List 1](#_Toc106700704)

[Related Documents 1](#_Toc106700705)

[1 Introduction 4](#_Toc106700706)

[1.1 Purpose 4](#_Toc106700707)

[1.2 Project Scope 4](#_Toc106700708)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc106700709)

[2 User Requirements Definition 5](#_Toc106700710)

[3 Requirements Specification 6](#_Toc106700711)

[3.1 Functional requirements 6](#_Toc106700712)

[3.1.1 Use Case Diagram 6](#_Toc106700713)

[3.1.2 Requirement 1: User Logs in 7](#_Toc106700714)

[3.1.3 Requirement 2: User Registers an account 8](#_Toc106700715)

[3.1.4 Requirement 3: User Views and Adds a Diary entry 10](#_Toc106700716)

[3.1.5 Requirement 4: View and Add GP visits 12](#_Toc106700717)

[3.1.6 Requirement 5: Analyse Diary entries 14](#_Toc106700718)

[3.2 Non-Functional Requirements 15](#_Toc106700719)

[3.2.1 Performance/Response time requirement 15](#_Toc106700720)

[3.2.2 Availability requirement 15](#_Toc106700721)

[3.2.3 Recover requirement 15](#_Toc106700722)

[3.2.4 Security requirement 15](#_Toc106700723)

[3.2.5 Reliability requirement 16](#_Toc106700724)

[3.2.6 Maintainability requirement 16](#_Toc106700725)

[3.2.7 Extendibility requirement 16](#_Toc106700726)

[4 Interface requirements 16](#_Toc106700727)

[4.1 GUI 16](#_Toc106700728)

[5 System Architecture 25](#_Toc106700729)

[6 System Evolution 25](#_Toc106700730)

[7 Bibliography 26](#_Toc106700731)

# Introduction

## Purpose

This document is to lay out the requirements for the development of this project also known as Migraine Diary. The main purpose of a migraine diary is to find the trigger or triggers of a user’s migraine and therefore allow the user to adjust their lifestyle in accordance with the trigger or triggers.

The target audience is for anyone who suffers from migraines. There is no restrictions on gender, race, age etc. The only restriction is whether a user suffers from migraines or not. The project is a web application in that it would provide an online resource for potential users without the reliance of apps on tablets or smartphones.

## Project Scope

The overall scope of this project is to create a web application that facilitates an organized diary to aid in the identification of a trigger or triggers of a user’s migraines.

The project requires the use of Laravel, a web application that uses PHP (definition below). Laravel allows the use of HTML, CSS, jQuery and JavaScript, all of which will be used alongside PHP. SQLite will be used to interact with a database by either querying, adding, deleting or altering data.

This project will take place over approximately nine weeks during semester three of this course. The IDE of choice will be Visual Studio Code which will be used to create the migraine diary web application.

The user can login or register an account when a migraine is triggered and navigate to the web page that contains the web form for the diary and provide information in regard to any food items, fluid intake, any medications take, stress and a “comment” field that allows for any extra comments in regard to the migraine. The user can then view that entry and, if relevant, past entries which will be sorted in chronological order with the first being the most recent.

The user can also, after logging in or registering, navigate to another web form for the GP tracking feature and provide information in regard to who the GP is, any medication changes and advice given. This allows the user to keep track of a user’s treatment for migraines.

The analysis web page takes the information a user has provided and provides data on the frequency of migraine occurrences in relations to each potential trigger. This can then provide indications as to any possible triggers that are relevant to the user.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Laravel | A PHP based web framework used to create web applications |
| PHP | Hypertext Preprocessor, a programming language that works on the server-side |
| HTML | Hyper Text Markup Language, a markup language to determine the structure of a website. |
| CSS | Cascading Style Sheets, a language that is used to style a website. |
| JavaScript | A client-side programming language that is used to create dynamic and interactive content |
| jQuery | A JavaScript library that makes the use of JavaScript easier |
| SQLite | A version of SQL that does not require a server to function (sqlite, no date). |
| IDE | Integrated Development Environment, an application that provides a development environment |
| GP | General Practitioner, a doctor |

# User Requirements Definition

In order for the user to be able to interact with the web application a user would need a device with a browser and an internet connection. There are no other requirements in terms of operating system or device or anything else. While the web application will be developed on a Windows 10 operating system on an Acer laptop, it should be viewable on any device with a browser and internet connection (the web application will not be deployed).

# Requirements Specification

## Functional requirements

This subsection describes the functionalities the web application will carry out:

1. The web application will be able to display information about migraines (with references).
2. The user will be able to log in or register.
3. The web application will be able to display the web form for diary entries.
4. The user will be able to view diary entries in chronological order with most recent displayed on top.
5. The web application will be able to display the web form for tracking GP visits.
6. The user will be able to view GP visit entries in chronological order with most recent displayed on top.
7. The web application will be able display a simple analysis of the diary entries to provide an indication of the possible triggers.

### Use Case Diagram

Diagram, schematic

Description automatically generated

### Requirement 1: User Logs in

#### Description & Priority

The user will need to enter a username and password to access the main features of the Migraine Diary web application.

#### Use Case

Diagram

Description automatically generated

**Scope**

The scope of this use case is to log in to the system

**Description**

This use case describes the user logging into the Migraine Diary web application

**Precondition**

1. The user has internet access
2. The user is using a browser
3. The user knows the username
4. The user knows the password

**Activation**

This use case starts when the user clicks on the login button

**Main flow**

1. The user clicks on the login button on the home page of the Migraine Diary web application
2. The system displays the login web form
3. The user fills in the username and password
4. The user clicks the sign in button
5. The system will display the page that will display diary entries

**Alternate flow**

Step 3 of main flow : The user gives wrong username and/or password

1. The system will display an error message
2. The user will be allowed to try again, no limits on the number of attempts

**Post condition**

The user is logged into the system

### Requirement 2: User Registers an account

#### Description & Priority

The user will create a username and password so as to create an account so as to access the main features of the Migraine Diary web application.

#### Use Case

Diagram

Description automatically generated

**Scope**

The scope of this use case is to register an account with the system

**Description**

This use case describes the user registering (creating) an account for the Migraine Diary web application

**Precondition**

1. The user has an internet connection
2. The user is using a browser

**Activation**

This use case starts when the user clicks on the register button

**Main flow**

1. The user clicks on the register button
2. The system displays the registration web form
3. The user fills in a username
4. The user fills in a password twice to confirm the password is correct
5. The user clicks on the create account button
6. The system will display the page that will display the diary entries

**Alternate flow**

Step 4 of main flow: The two password entries do not match

1. The system will display an error message
2. The user proceeds to step 3

**Post condition**

The user has created an account and is logged into the system

### Requirement 3: User Views and Adds a Diary entry

#### Description & Priority

The user can then view diary entries and can add new entries

#### Use Case

Diagram

Description automatically generated

**Scope**

The scope of this use case is to view and possibly add a diary entry

**Description**

This use case describes the user viewing past Migraine Diary entries and to add a new entry

**Precondition**

1. The user has an internet connection
2. The user is using a browser
3. The user has logged into an existing account

**Activation**

This use case starts when the user has finished logging into the system

**Main flow**

1. The system directs the user to the page that displays the diary entries after logging in
2. The user can view past entries
3. The user clicks on the add entry button
4. The system will display the diary web form
5. The user fills out the necessary fields
6. The user will click on the submit button
7. The system will validate the data provided by the user
8. The system will direct to the page that will display the diary entries with the new addition.

**Alternate flow**

Step 6 of main flow: The user enters wrong or omits required information (no date given)

1. The system will display an error message
2. The user proceeds to step 3

**Post condition**

The user has viewed past entries and added a new entry

### Requirement 4: View and Add GP visits

#### Description & Priority

The user can view records of previous GP visits and add new entries

#### Use Case

Diagram

Description automatically generated

**Scope**

The scope of this use case is to view past GP visits and possibly add a new GP visit

**Description**

This use case describes the user viewing past GP visit entries and possibly to add a new entry

**Precondition**

1. The user has an internet connection
2. The user is using a browser
3. The user has logged into an existing account

**Activation**

This use case starts when the user has finished logging into the system and has navigated to the page that displays past GP visits

**Main flow**

1. The system directs the user to the page that displays the diary entries after logging in
2. The user then navigates to the GP visits display page
3. The system will display the page that shows past GP visits
4. The user clicks on the add GP visit button
5. The system will display the GP visit web form
6. The user fills out the necessary fields
7. The user will click on the submit button
8. The system will validate the data provided by the user
9. The system will direct to the page that will display the past GP visits page with the new addition.

**Alternate flow**

Step 6 of main flow : The user enters wrong or omits required information

1. The system will display an error message
2. The user proceeds to step 5

**Post condition**

The user has viewed past entries and added a new entry

### Requirement 5: Analyse Diary entries

#### Description & Priority

The user can view the results of the analysis of the Migraine Diary entry to give indications of potential triggers

#### Use Case

Chart, bubble chart

Description automatically generated

**Scope**

The scope of this use case is to view the results of the analysis of the migraine diary entries

**Description**

This use case describes the user viewing the results of the analysis of the migraine diary entries

**Precondition**

1. The user has an internet connection
2. The user is using a browser
3. The user has logged into an existing account
4. The user having at least one diary entry

**Activation**

This use case starts when the user has finished logging into the system and has navigated to the analysis page

**Main flow**

1. The system directs the user to the page that displays the diary entries after logging in
2. The user then navigates to the analysis page
3. The system will display the results of the analysis of the migraine diary entries
4. The user views the results

**Post condition**

The user gets indications of the potential triggers relevant to the user.

## Non-Functional Requirements

### Performance/Response time requirement

The Migraine Diary web applications performance shall be reliable in that the diary and GP visit entries shall be displayed correctly, and the speed shall be relatively quick, emphasis will be more on performance than speed as the information provides useful hints as to what the persons trigger or triggers will be, therefore reliable delivery will be needed.

### Availability requirement

This web application will not be deployed during this semester to avoid any GDPR and ethical issues. Instructions on how to install Laravel and the dependencies will be given.

### Recover requirement

For this Migraine Diary project, only fake data will be used to help test the functionality of the web application therefore no data back-up will be undertaken other than creating a back-up copy of the whole project as SQLite will be the query language used.

### Security requirement

User authentication will be set up so as to ensure that the user can only view the users entries and analysis. User input will be validated to ensure that the correct data is input and no injection attacks can be inserted.

### Reliability requirement

The Migraine Diary web application shall be able to carry out its function continuously as long as there is an internet connection.

### Maintainability requirement

The code shall be kept as clear and easy to read as possible with comments and any conventions followed as much as possible to increase the ease of adding any potential updates. This should decrease the time for the Migraine Diary web application to be offline when the updates are applied.

### Extendibility requirement

The purpose of this Migraine Diary web application is to aid in the users search of finding any triggers for the users’ migraines. At the moment, there is not much room for adding any more features, but this is subject to change.

# Interface requirements

## GUI

The interface that will be used by the Migraine Diary web application for this project will use a Graphical User Interface (GUI). A GUI is part of an application that a user interacts with to perform actions with the use of icons and other visual indicators (Stoltzfus, 2021).

The top menu or navigation will have different links depending on whether a user is logged in or not. If the user is not logged, the navigation would have three links – Home, Log In, Register. If the user is logged in the navigation would have five links – Home, Diary, GP Tracker, Analysis, Sign Out.

If the user has clicked on the log in page and has not registered an account, the log in form will be directly below the navigation. The registration form on the register page will be the same. Therefore, there is no need to have more links between the two web pages. After the user has logged in or registered, the user will be directed to the Diary page with the navigation described above with the five links.

The Diary page will contain a button that links to the page that adds a new Diary entry. The GP Tracker page will also contain a button that links to the page that adds a new GP visit entry. When the user signs out by clicking Sign Out, the user will be directed to the Home Page with the navigation back to having three links.

The gernal layouts are shown below (next page).

**Home**

A piece of paper with writing on it

Description automatically generated

**Log In**

A piece of paper with writing on it

Description automatically generated

**Register**

A piece of paper with writing on it

Description automatically generated

**Diary**

A piece of paper with writing on it

Description automatically generated

**GP Tracker**

A piece of paper with writing on it

Description automatically generated

**Analysis**

Text, letter

Description automatically generated

**Add Diary Entry**

Text, letter

Description automatically generated

**Add GP Visit Entry**

A piece of paper with writing on it

Description automatically generated

# System Architecture

Chart, diagram, box and whisker chart

Description automatically generated

This architecture will allow the user to easily create an account or log in to an existing account and then view or add diary entries or GP visits following the use of CRUD (Create, Read, Update, Delete). The purpose of following the use of CRUD is to give the user a strong level of control over the flow of information provided.

# System Evolution

This web application is to create an online migraine diary to aid in the identification of possible migraine triggers. This web application could be used as a template for determining what can be causing an allergic reaction or as a template for other medical conditions.

# Bibliography

SQLite, (no date) ‘SQLite Is Serverless’, SQLite. Available at: <https://www.sqlite.org/serverless.html> [Accessed 23/06/2022].

Stoltzfus, J. (2021) ‘Graphical User Interface (GUI)’, Techopedia, 28 May. Available at: <https://www.techopedia.com/definition/5435/graphical-user-interface-gui> [Accessed 18/06/2022].