



Migraine Tracking Software

Client: Eliza Brock

This is the full requirements and specifications document for our Migraine Tracking Web Application.

Brian Baro, Matt Klaasse, Jackson Melling, Mark Wlodarski
11/8/2011

Table of Contents

Executive Summary..... 7

Introduction 8

Client Background..... 9

Current System 10

Users/Stakeholders..... 11

 Users 11

 Client/Maintenance Team 11

 Medical Professionals 11

User Need 12

Alternatives and Competition..... 13

 Pen and Paper 13

 Ubiqi Health 13

 Other Software Solutions..... 13

Product Overview 14

 Product Perspective 14

 Elevator Statement 14

 Summary of Capabilities 14

 Assumptions..... 14

 Rough Estimate of Cost..... 14

Features 15

 Migraine Tracking System..... 15

 Long Term Tracking..... 15

 Data Storage..... 15

 User Accounts 16

 Security 16

 Exportable/Printable..... 16

 Analysis of Migraine Data 16

 Editable Data..... 17

 Mobile Friendly 17

Solution Constraints..... 18

User Profiles..... 19

Occasional Migraine Sufferer.....	19
Moderately Frequent Migraine Sufferer	19
Frequent Migraine Sufferer	19
Use Cases	20
User Creates an Account.....	20
Description	20
Pre Conditions.....	20
Basic Flow.....	20
Alternate Flow 'A': User Cancels Account Creation.....	20
Alternate Flow 'B': Invalid Account Information	20
Alternate Flow 'C': Database is Currently Not Operational	20
Post Conditions	20
User Changes Account Preferences.....	21
Description	21
Pre-Conditions	21
Basic Flow.....	21
Alternate Flow 'A': User Cancels Preferences Changes	21
Alternate Flow 'B': Data Entered is Invalid	21
Post Conditions	21
User Authentication.....	22
Description	22
Pre Conditions.....	22
Basic Flow.....	22
Alternate Flow 'A': User Enter Incorrect Username or Password	22
Post Conditions	22
User Accesses/Edits Information about a Previous Report	23
Description	23
Pre Conditions.....	23
Basic Flow.....	23
Alternate Flow 'A': User Presses Cancel Button on View Report Page.....	23
Alternate Flow 'B': User Inputs Invalid Data	23
Post Conditions	23

User Exports Data to a PDF File	24
Description	24
Pre Conditions.....	24
Basic Flow.....	24
Alternate Flow 'A': User Cancels the Operations.....	24
Alternate Flow 'B': User Presses Preview PDF Button	24
Post Conditions	24
User Views Analytics about their Migraine Data	25
Description	25
Pre Conditions.....	25
Basic Flow.....	25
Alternate Flow 'A': User Cancels the Operations.....	25
Alternate Flow 'B': User Cancels the Operations.....	25
Post Conditions	25
User Logs Out of Website	26
Description	26
Pre Conditions.....	26
Basic Flow.....	26
Post Conditions	26
User Deletes Their Account	27
Description	27
Pre Conditions.....	27
Basic Flow.....	27
Alternate Flow 'A': User Stops the Deletion Process	27
Post Conditions	27
User Submits a Report	28
Description	28
Pre Conditions.....	28
Basic Flow.....	28
Alternate Flow 'A': User Cancels the Operation	28
Alternate Flow 'B': The Information That the User Entered is Invalid.....	28
Post Conditions	28

Data Flow Diagram.....	30
Level 0	30
Level 1	31
Level 2	34
Non-Functional Requirements.....	39
Usability Requirements.....	40
Performance Requirements.....	41
Reliability Requirements.....	41
Supportability Requirements.....	41
Documentation	42
Readme File.....	42
Installation	42
Installation Process	42
Legal	42
Terms and Conditions	42
Licensing Requirements.....	42
All Resources Used Will be Licensed.....	42
Hardware Software Interface	43
User Interface	44
Coding Standards	45
Change Control	45
Test Cases.....	Error! Bookmark not defined.
Account Creation	Error! Bookmark not defined.
User Authentication.....	Error! Bookmark not defined.
Changing Account Preferences	Error! Bookmark not defined.
User is Able to Submit Valid Current Migraine Report	Error! Bookmark not defined.
User is Able to Edit Previous Report	Error! Bookmark not defined.
User is Able to Export to PDF	Error! Bookmark not defined.
Logout	Error! Bookmark not defined.
Delete Account.....	Error! Bookmark not defined.
User Submits Reports.....	Error! Bookmark not defined.
Usability Report	55

Process	56
Analysis	57
Findings	58
Buttons on Homepage	58
Change How Submit Works	58
Export.....	58
Log out	58
Interaction Architecture	59
Initial Interface Design	61
Revised Interface Design.....	61
Glossary.....	62
References	63
Change Log.....	64

Executive Summary

We are developing a Migraine Tracking system for Eliza Brock of Eliza Brock Software. This program will be a web based application for ease of use to the average user, with a possible mobile friendly version to be developed in the future. By request of the client, we are using the Ruby on Rails development framework. The client has also given a specified database management tool. The software must also be usable by an individual currently suffering from a migraine because of the unique purpose of our software. Our client also wishes that the software be made easy for her to maintain and update after we deploy the software at the end of the project. Because the end goal of the product is to report migraine data to the user for use by their doctors, the software will also be able to generate a printable report of the data.

This document contains background information on the current situations, such as client backgrounds, user needs, current solutions, and a list of confirmed features. Additionally it contains use cases, storyboards, non-functional requirements, feature mapping, constraints, prototypes, DFDs, change control plan, testing plan, Ruby coding standards that we will be using, usability study details, an interface prototype both before and after changes made based on the usability tests, and an interaction architecture.

Introduction

This document contains every bit of documentation that has gone in to the development of the Migraine Web Application. This documentation includes in part the features of the product, the use cases that have been developed and the test cases that were based on them. Also included is the usability tests and there results conducted in order to determine the user friendliness of the application. Included as well is some initial back end designs such as a data flow diagram. All of this has been used to develop an initial prototype used in usability testing, the results of which are also included in this document.

Client Background

Our client, Eliza Brock, is the head of a start up software company called Eliza Brock Software. She is an expert in Ruby on Rails and comes from a family of migraine sufferers. She intends to market the software as well as use it herself. During the requirements phase she will be involved in helping decide what requirements are of the highest priority to the success of the project. She will also have access to the source code through our team's Github repository which she will provide. After we deliver our final deliverable she will take full responsibility and control of the project including updates, marketing, and software maintenance.

Current System

Currently the primary method of migraine tracking is through a pen and paper migraine journal containing information on migraine onset, triggers, and any other relevant information dictated by a medical professional. This presents some difficulties including having easy access to a journal while having a migraine, forming coherent thought to write down the relevant information, and legibility. A software solution to migraine tracking would make tracking easier to do by providing fields to fill in relevant information rather than relying on the user to determine what is relevant while suffering a migraine. It also makes it easier to keep track of their data because the information is stored on a database rather than an easily lost journal.

Currently our client does not have a software solution for migraine tracking, however, several migraine tracking software solutions do exist which will serve as competition to our solution.

Users/Stakeholders

Users

Migraine sufferers are the main users of the system. They are a subset of the overall population that is a reflection of the overall population. Thus they have average technical knowledge and average medical knowledge. The only thing they all have in common is their shared suffering of migraines. When suffering from a migraine, the use of software becomes more difficult, and thus a graphic interface that allows for easy and pleasant reporting of data is required.

Their environment is usually themselves, although occasionally there could be a non migraine sufferer. In that case the non migraine sufferer could aid them in the use of the system. They could be anywhere, as long as there is internet they should be able to use the system.

Client/Maintenance Team

The client and by association, the maintenance team are highly fluent with computers, and skilled in the use of Ruby on Rails. Our client will take over all responsibility for the project after completion, and will complete any updates and marketing deliverables for the project. She requires well documented code that is easy to understand. It is also required that the software product be usable and marketable, and complies with all relevant standards and regulations.

The client will be choosing the majority of her, and her maintenance team's environment. She will be choosing the development environment and the size of her team. She currently does not use any other applications for migraine sufferers.

Medical Professionals

Medical professionals are stakeholders, and by definition they have strong medical backgrounds. The ones involved with this system will also be highly educated on the subject of migraines. They will require reports generated by the system to produce medical solutions for the migraine sufferers. These stakeholders will need the reports to be well laid out, and for all the information tracked to be well documented. Our system's definition of success is that medical professionals will be able to properly treat the users with the reports generated by the system.

User Need

Need: Ability to record information about individual migraines

Affects: Users

Solution: Software accepts input from user about their migraine and stored the data on an external database

How it will address the problem: Allows the user to easily record their information in a location that they will always have access to

Need: Information to diagnose migraine symptoms

Affects: Users, Medical Professionals

Solution: User can print a report of their migraine history to give to a medical professional

How it will address the problem: Medical professionals can use the migraine information in their diagnosis

Need: Software must be maintainable

Affects: Maintenance Team

Solution: Ensure that our code is well documented and easily expandable

How it will address the problem: Cleaner and more expandable code will give the maintenance team an easier job with future maintenance and releases of the software

Need: A recording solution that is usable during a migraine

Affects: User

Solution: Use a migraine friendly color scheme with low contrast.

How it will address the problem: Will allow people using the software to experience as little discomfort as is possible

Need: The ability to record migraines from anywhere

Affects: User

Solution: Produce a mobile friendly version of the web page

How it will address the problem: A mobile friendly version will allow any user with a migraine to easily record details using their phone

Alternatives and Competition

Pen and Paper

The pen and paper solution is currently the most commonly recommended solution by medical professionals. It involves the user keeping a paper journal of their migraine symptoms, triggers, and whatever else their doctor deems appropriate.

The advantages of this solution are that it is portable, cheap, and is easy to use while experiencing a migraine.

The disadvantages of this solution are that the user must filter what information is relevant while experiencing the migraine, it is easy to misplace or lose the journal, and you must keep the journal with you wherever you go.

Ubiqi Health

Ubiqi Health is currently the most commonly used migraine tracking software. They have a web app along with mobile apps for all major smart phones. The software handles tracking of common triggers, treatments, and migraine onsets and uses that information to track trends to help the user.

The advantages of this solution are that it can be used from anywhere with access to a smart phone, the ability to track trends, and its ease of use.

The primary disadvantages of the software are that it does not have a migraine friendly UI, and is not fun for the user to use

Other Software Solutions

Several other software solutions exist for tracking migraines however many of them have the same disadvantages as Ubiqi. Many of them also do not have mobile apps to support use of their software on the move.

Product Overview

Product Perspective

The product is a stand-alone web based application written in Ruby on Rails. The goal of this project is to be of similar ease of use to a pen and paper journal with the accessibility of a web application. Using a web app will be much faster for the user than keeping a pen and paper journal because rather than having to fill in all the information, they simply need to fill out a form with their personal symptoms. In addition, the software will be cheaper for the user because they will not need to meet the material costs of a paper journal.

Elevator Statement

Currently there are no intuitive and user friendly web based migraine tracking applications. We are seeking to design a web based application to aid migraine sufferers in tracking any data relevant to their condition for use by medical professionals in treatment of their condition. Our goal is to design an application that is comparable to a pen and paper system with the easy accessibility of a web based application.

Summary of Capabilities

Customer Benefit	Supporting Features
Lower migraine pain	migraine friendly, ease of use
easily accessible	mobile friendly, exportable/printable, data storage, ease of use
access to data history	exportable, data storage, editable data, user accounts, statistical analysis of migraine data, migraine statistics
fun to use	mobile friendly, fun, migraine friendly, ease of use
Expandable/Maintainable	maintainable, expandable, localized, secure
Help migraine treatment	track migraine statistics, exportable/printable, data storage, editable data
Secure Data	Regulations, secure, user accounts

Assumptions

- Ruby is supported by all major web browsers
- the client is capable of hosting our software
- Assuming that there exists a color scheme for our software that is usable by those suffering migraines
- Assuming that users will want/have access to a computer while suffering a migraine
- Assuming that we are capable of completing the software within our time frame
- Assuming that the client will accept our software solution as a success

Rough Estimate of Cost

Monetary cost: \$0.00

Time Cost: Approximately 30 Weeks

Man Power: Four Developers, One Project Manager, One Project Supervisor

Legend
Critical: Of the utmost importance
High: Important
Medium: Moderately Important
Low: Not Important

Features

Note: Current status for all features is currently “Approved”

Migraine Tracking System

ID: #1

Priority: Critical

Risk: Medium

Stability: High Stability

Owner: Everyone

Target Release: 1.1

Reason: This is the main feature of the product. Without this, our project is an automatic failure. This will be scheduled for an early release, and the entire group will be responsible for working on this part. Medium risk because if it fails, the product is a failure, but the difficulty will not be high. We do not expect this feature to change.

Long Term Tracking

ID: #13

Priority: Critical

Risk: Medium

Stability: High Stability

Owner: Everyone

Target Release: 1.1

Reason: To separate our product from competing migraine tracking systems, our system will be capable of tracking not only migraines but also the times and amounts of different medications that the user is taking to help to find patterns between when the user takes their medication and their migraine onset. This will be possible even when the user has not yet experienced a migraine as a result of the medication or otherwise.

Data Storage

ID: #2

Priority: Critical

Risk: Medium

Stability: Medium

Owner: Everyone

Target Release: 1.1

Reason: This will allow users to gather and use information about their past migraines which is necessary for our project success. The gathering and storage of data will likely not change during the development, and will need to be implemented early.

User Accounts

ID: #4

Priority: Critical

Risk: High

Stability: Medium

Owner: Everyone

Target Release: 1.4

Reason: User Accounts is an important feature the needs to be implemented so that each user can access only their own data. Failure to correctly implement this could create security holes that could be taken advantage on to find other users data. The method that our client wants us to use could change over the course of the project.

Security

ID: #5

Priority: Critical

Risk: High

Stability: Low

Owner: Everyone

Target Release: 1.4

Reason: Security will be the implemented along with #5, and will include secure logins, logouts, and data transfer. The user must feel that their data is secure so that they will be comfortable entrusting the website with their personal information.

Exportable/Printable

ID: #6

Priority: High

Risk: Low

Stability: High

Owner: Everyone

Target Release: 1.6

Reason: The data that the user inputs needs to be able to be exported into a viewable format that the user can show to a medical professional. It is not likely to change since the client requested this feature.

Analysis of Migraine Data

ID: #9

Priority: High/Medium

Risk: Low

Stability: Low

Owner: Everyone

Target Release: 1.6

Reason: This feature will be released along with exportable data. This could become a more important feature in the future, but should be easy for the team based on the back end oriented nature of a majority of the team.

Editable Data

ID: #10

Priority: Medium

Risk: Medium

Stability: High

Owner: Everyone

Target Release: 1.2

Reason: This will be released following the addition of the ability to enter and view data in the database. We do not expect this feature to change.

Mobile Friendly

ID: #12

Priority: Low

Risk: Low

Stability: Low

Owner: Everyone

Target Release: 2.1

Reason: A mobile friendly version of the program would be a nice feature, but might not be within the scope of the project. It could change based on client needs.

Solution Constraints

One of the most critical constraints is that the system must meet all relevant medical regulations and standards. This will be an issue as user medical data will be stored indefinitely if they use this system.

One of our constraints is that the project must be done using Ruby on Rails with Cucumber Testing Framework. The rationale for this constraint is that the client requested we use these tools, as she considers them to be the most effective and is highly familiar with them. For the same reasons we are required to use a database backed by Heroku. Another constraint is that the system must be a web application, with the rationale behind this constraint being that the user's data should not be stored locally and that the user can utilize the system anywhere there is internet.

It is also required that we use GitHub for our source control. The reason behind this is that it allows the client to access the source code for any reason she desires.

The client has specified that because we will be using Ruby on Rails, proper coding documentation is not a priority, however, a well formatted and complete README file is a priority for the benefit of anyone that needs to work on the project in the future.

The final constraint is that the project must be completed by the end of this current academic year.

User Profiles

Occasional Migraine Sufferer

An occasional migraine sufferer is a user who experiences migraines at a rate of approximately once per month or less frequently. These users will likely not access our software as often as moderately frequent or frequent migraine sufferers and will therefore not submit data as often as other users. This will cause their data to be less intensive than other user's data.

Moderately Frequent Migraine Sufferer

A moderately frequent migraine sufferer is someone who experiences migraines more than once a month on average but still experiences it less than once per week. Their data will be more accurate than occasional migraine sufferers because they will access the software more frequently. They are more likely to enter data even when not experiencing migraines because of the frequency of their migraines.

Frequent Migraine Sufferer

A frequent migraine sufferer is a user that has migraines once weekly or more. These users will have the most detailed information in our system because of the frequency of their migraines. They are also more likely to use the software to its fullest extent because of the severity of their condition and are more likely to track all the medication usage and other possible migraine triggers outside of when they are experiencing migraines.

Use Cases

User Creates an Account

ID#: UC1

Description

The user uses the account creation page to create an account as well as enter some personal information.

Pre Conditions

- 1) User is on our website
- 2) Our website is currently operational

Basic Flow

- 1) User selects option to create a new account
- 2) User enters information for their account [See Alt Flow 'A']
 - a. Email Address
 - b. Password
- 3) User Presses Submit
- 4) Information sent to database for confirmation and to be written [See Alt Flows 'B' & 'C']
- 5) User gets confirmation of account creation
- 6) Email sent to user email address confirming account creation
- 7) User sent to home screen

Alternate Flow 'A': User Cancels Account Creation

- 1) During step 2 of the Basic Flow user presses cancel button
- 2) User redirected to login page

Alternate Flow 'B': Invalid Account Information

- 1) User enters invalid information
- 2) User presented with error message with specific information about what information was invalid
- 3) User redirected to step 2 of Basic Flow

Alternate Flow 'C': Database is Currently Not Operational

- 1) After Step 4 an error is displayed that states that the database is down and to try again later
- 2) User redirected to home page

Post Conditions

- 1) User has an account
- 2) User is at the home screen

User Changes Account Preferences

ID#: UC2

Description

User uses the account editing page to change details of their account such as password and age.

Pre-Conditions

- 1) User is logged in
- 2) User is on home page

Basic Flow

- 1) User presses account preferences button
- 2) User changes the values of the fields that they would like to change [See Alt Flows 'A' & 'B']
- 3) User presses save changes button
- 4) System sends change requests to database
- 5) User gets confirmation of preference changes
- 6) User is sent to home page

Alternate Flow 'A': User Cancels Preferences Changes

- 1) User presses cancel during step 2 of the Basic Flow
- 2) User returned to home screen

Alternate Flow 'B': Data Entered is Invalid

- 1) User enters invalid information
- 2) User presented with error message with specific information about what information was invalid
- 3) User redirected to step 2 of Basic Flow

Post Conditions

- 1) Any preferences that the user wishes to change have been changed
- 2) The user is on the home page

User Authentication

ID#: UC3

Description

User logs in to the system using their account name and password

Pre Conditions

- 1) The user is on our site
- 2) The user is not currently logged in

Basic Flow

- 1) User enters username and password
- 2) User presses login button or enter key
- 3) System checks username and password [see Alt Flow 'A']
- 4) User is logged in

Alternate Flow 'A': User Enter Incorrect Username or Password

- 1) The system is unable to find a valid username password combination in the system
- 2) The user is redirected to the login screen and begins step 1 again

Post Conditions

- 1) The user is logged in

User Accesses/Edits Information about a Previous Report

ID#: UC7

Description

User wants to view details about or change details in a previous report.

Pre Conditions

- 1) User is logged in
- 2) User is on home page

Basic Flow

- 1) User presses View/Edit Previous Reports Button
- 2) User is presented with a calendar containing all previous reports
- 3) User selects report that they wish to view or edit
- 4) User is presented with a form that allows them to see and edit data on that report [See Alf Flow 'A']
- 5) User presses save button to save any changes
- 6) Changed report details submitted to database [See Alt Flow 'B']
- 7) User is returned to the calendar page [See Step 2 of Basic Flow]
- 8) User presses home button to return to home page

Alternate Flow 'A': User Presses Cancel Button on View Report Page

- 1) User presses cancel button
- 2) Return to step 7 of basic flow

Alternate Flow 'B': User Inputs Invalid Data

- 1) User enters invalid information
- 2) User presented with error message with specific information about what information was invalid
- 3) User redirected to step 4 of Basic Flow

Post Conditions

- 1) User has viewed/edited any reports that they wanted to edit
- 2) User is on home page

User Exports Data to a PDF File

ID#: UC8

Description

The user wants to receive a PDF file detailing their migraine information so that they can either print it out or just have a printable report.

Pre Conditions

- 1) User is logged in
- 2) User is on home page

Basic Flow

- 1) User selects Export Data button
- 2) User is presented with a list of options for the export
- 3) User selects desired options [See Alt Flow 'A']
- 4) User presses export button [See Alt Flow 'B']
- 5) PDF download started through user's web browser
- 6) User redirected to home page

Alternate Flow 'A': User Cancels the Operations

- 1) User presses cancel button
- 2) User is redirected to the home page

Alternate Flow 'B': User Presses Preview PDF Button

- 1) User presses Preview rather than export button
- 2) User is shown PDF file in their web browser with the option to save the file

Post Conditions

- 1) User has downloaded a PDF of their migraine data
- 2) User is on the home page

User Views Analytics about their Migraine Data

ID#: UC9

Description

The user wants to view analytic data about their migraines. They can do so using our analytic tracking feature.

Pre Conditions

- 1) User is logged in
- 2) User is on home page

Basic Flow

- 1) User selects View Analytics button
- 2) User is presented with a list of options for Data
- 3) User selects desired options [See Alt Flow 'A']
- 4) User presses Analyze button
- 5) User is shown a page with their desired analytic data [See Alt Flow 'B']
- 6) User presses the Done button
- 7) User redirected to home page

Alternate Flow 'A': User Cancels the Operations

- 1) User presses cancel button
- 2) User is redirected to the home page

Alternate Flow 'B': User Cancels the Operations

- 1) User presses change options button
- 2) Go to step 2 of Basic Flow

Post Conditions

- 1) User has viewed their analytics
- 2) User is on the home page

User Logs Out of Website

ID#: UC10

Description

The user logs out of the system, preventing them from accessing data only available when logged in

Pre Conditions

- 1) User is logged in
- 2) User is on home page

Basic Flow

- 1) User presses Log Off button
- 2) User is logged off

Post Conditions

- 1) User is no longer logged on
- 2) User is on the login screen

User Deletes Their Account

ID#: UC11

Description

The user deletes their account and confirms the deletion by entering their password. Then the system removes the account information from the database.

Pre Conditions

- 1) User is logged in
- 2) User is on the home page

Basic Flow

- 1) User presses the Account Preferences button
- 2) User presses the delete account button
- 3) User is presented with a confirmation dialogue that this will delete their account from the system and given the option to cancel
- 4) User presses continue button [See Alt Flow 'A']
- 5) User is presented with another confirmation dialogue that their data will not be recoverable and asked to either type the password to continue or cancel the process
- 6) User presses continue button [See Alt Flow 'A']
- 7) System removes account details and all reports by that account in the system and logs the deletion
- 8) User is redirected to login screen

Alternate Flow 'A': User Stops the Deletion Process

- 1) User presses the cancel deletion button on either of the confirmation dialogues
- 2) User is redirected to their home page and the deletion is canceled

Post Conditions

- 1) User no longer has an account on the database
- 2) User is on the log in screen

User Submits a Report

ID#: UC12

Description

This use case will consolidate the activity report submission and the migraine report submission into one easy reporting process. The user selects one type of report, then enters information about it, and is returned to the report submission page.

Pre Conditions

- 1) The user is logged in
- 2) The user is on the home page

Basic Flow

- 1) The user presses the submit reports button
- 2) The user is brought to a screen to select the amount of each report to submit
- 3) User selects which type of report they wish to submit.
- 4) User is taken to that report page
- 5) User enters information about the report [See Alt Flow 'A']
- 6) System records the data as a new report [See Alt Flow 'B']
- 7) User is returned to the report submission page.

Alternate Flow 'A': User Cancels the Operation

- 1) User presses cancel button
- 2) User is redirected to the report submission page.

Alternate Flow 'B': The Information That the User Entered is Invalid

- 1) The information entered by the user is invalid
- 2) The user is brought back to the same page with an error message explaining what they must change (step 5)

Post Conditions

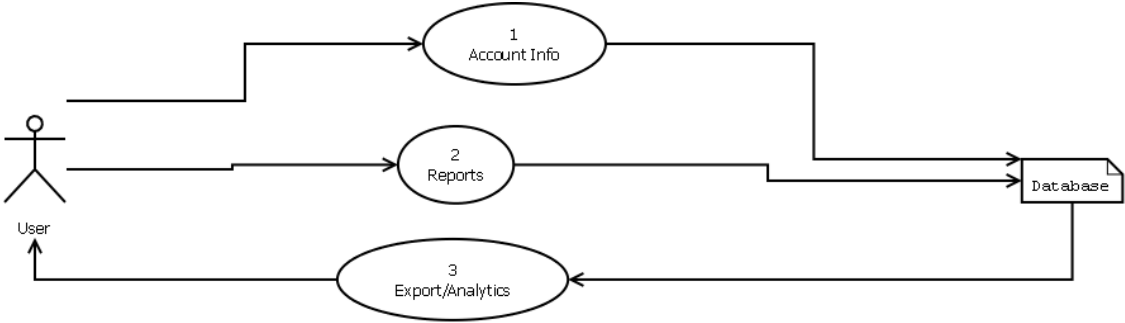
- 1) All reports that the user wanted to submit are submitted to the database
- 2) The user is on the report submission page

3) Feature Mapping

Use Case	Mapped Features
User Creates an Account	User Accounts (4)
User Changes Account Preferences	Editable Data (10)
User Authentication	Security (5)
User Enters Information about a Current Migraine	Migraine Tracking (1), Data Storage (2)
User Accesses/Edits Information about a previous report	Editable Data (10)
User Exports Data to a PDF File	Exportable/Printable (6)
User Views Analytics about their Migraine Data	Analysis of Migraine Data (9)
User Logs Out of Website	User Accounts (4)
User Deletes their Account	Editable Data (10), Security (5)
User Submits a Report	Long Term Tracking (13), Data Storage (2), Migraine Tracking (1)

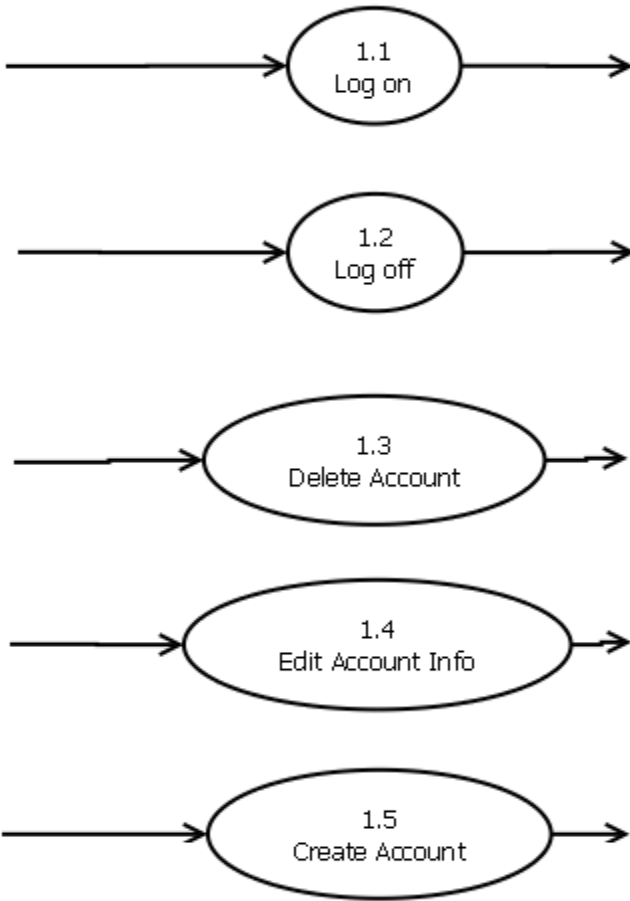
Data Flow Diagram

Level 0

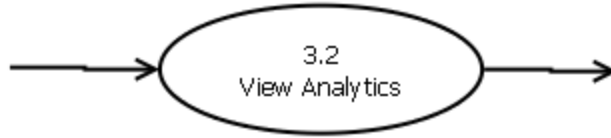
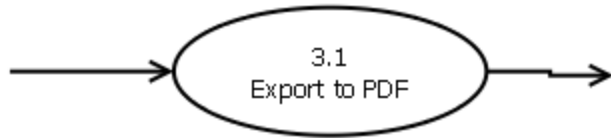


Each bubble is a grouping of a subset of features.

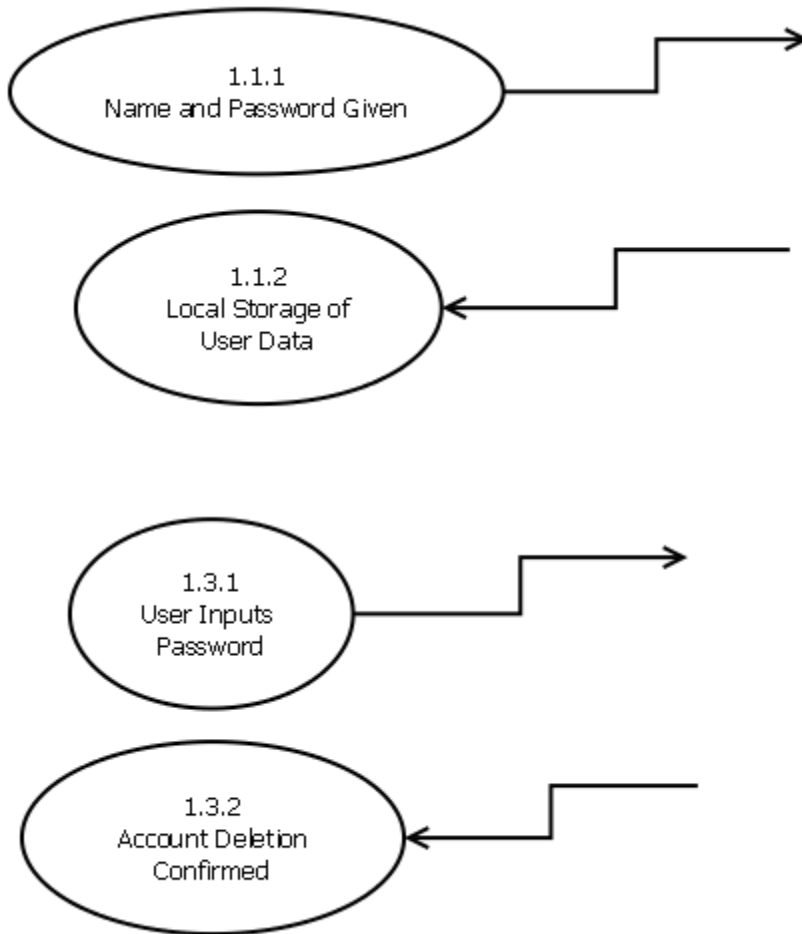
Level 1

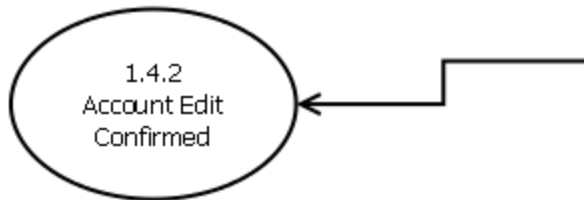


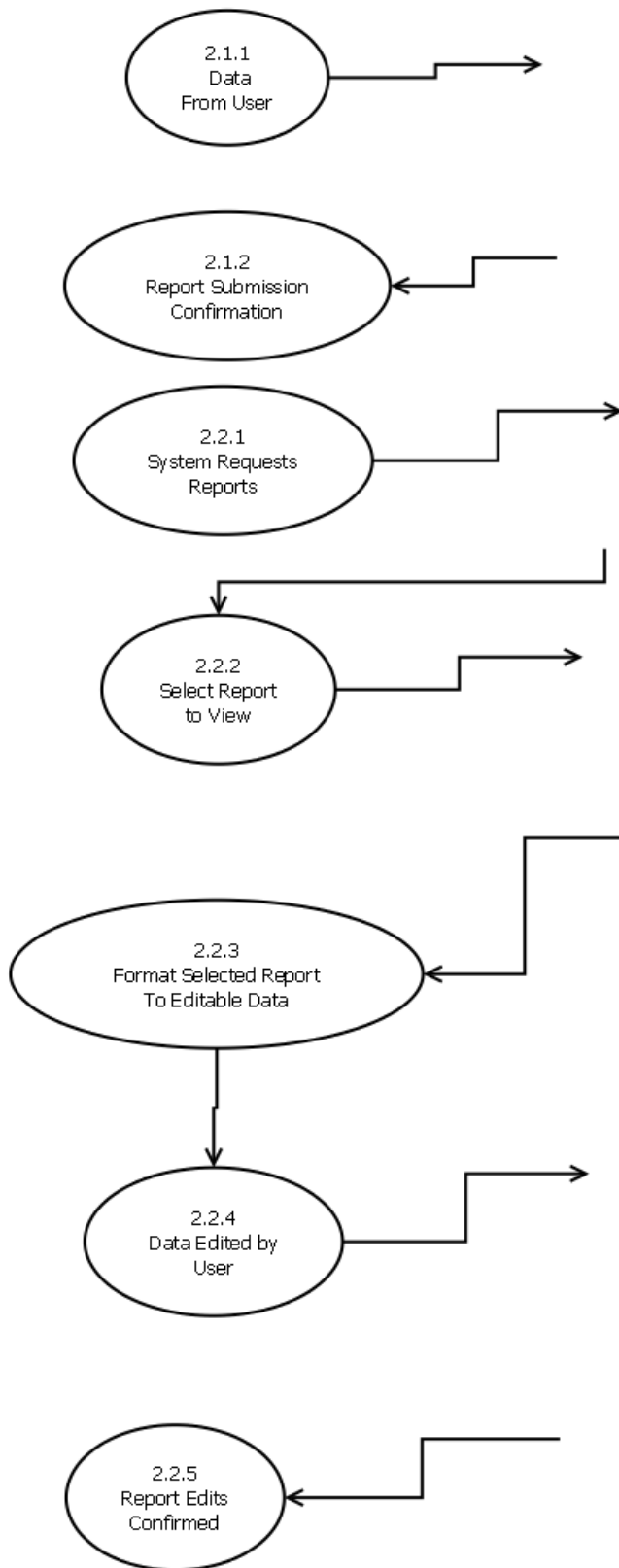


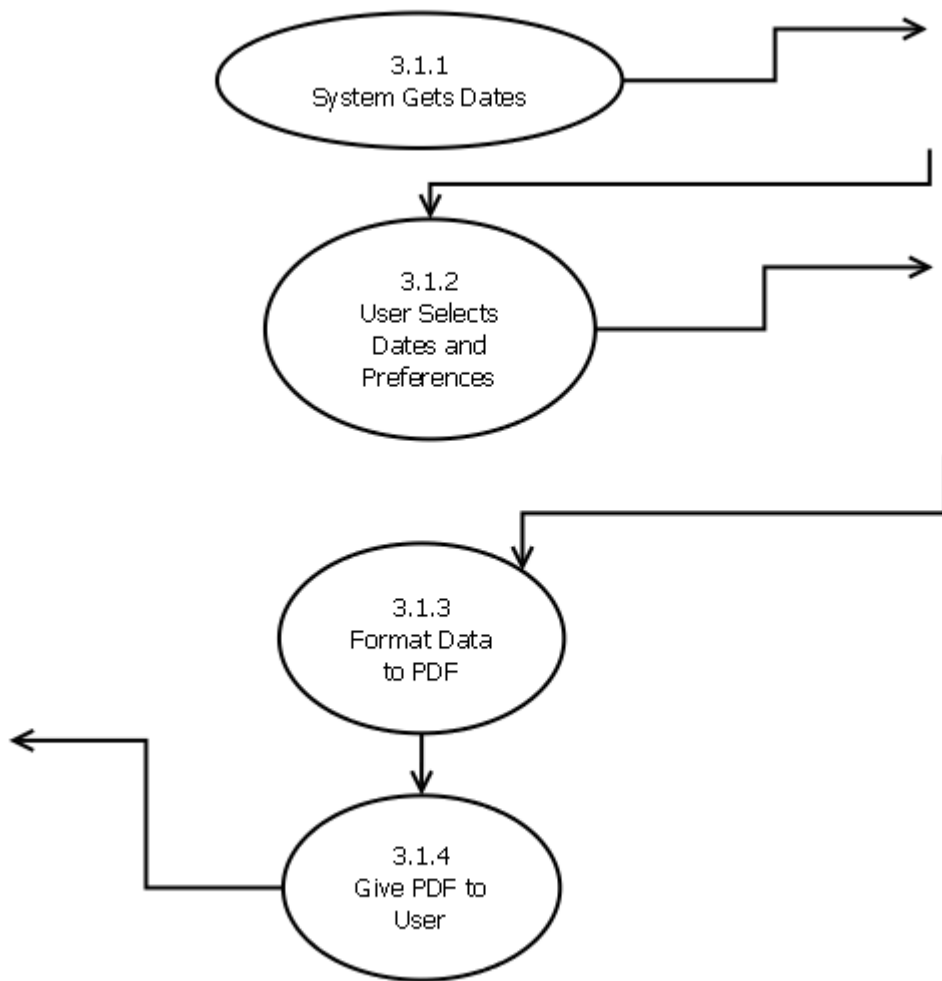


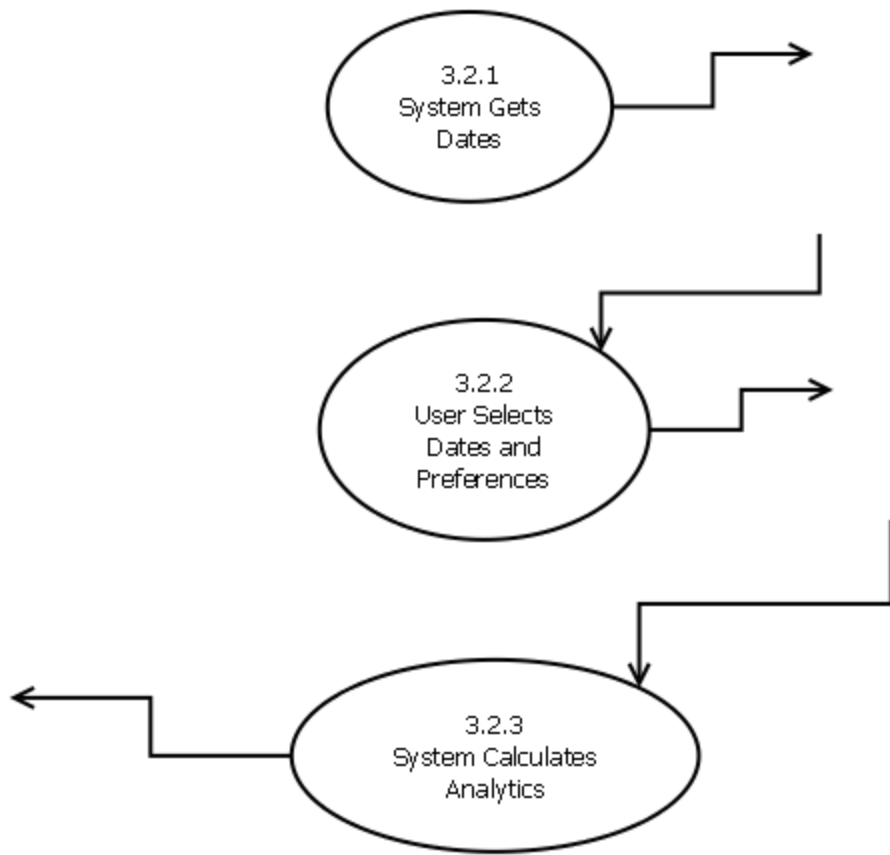
Level 2











Non-Functional Requirements

These requirements are requirements that are not tied directly to features or use cases but are still integral to the success of the project. Requirements for usability will cover requirements that effect the user experience, such a tool tips and how users feel about the program in comparison to other similar system. Performance requirements will cover how the system reacts to events, and how long it takes to return data. Reliability requirements are requirements that deal with how often the system is down, and what happens when it does go down. Supportability requirements deal with issues such as the README and help menus. Legal, documentation, licensing, and installation requirements will cover other issues not covered in the previous sections.

Usability Requirements

Tooltips on First Use

On first use of any given feature of our software, tooltips will be displayed next to each field giving instructions on the use of that field, and will continue to do so on each successive use until the user disables them.

User Privacy

All data stored on our software's databases will only be available to the user that provided the information and will never be sold or shared with anyone else.

At Least as User Friendly as Competition

Our software should be at least as user friendly and easy to use as Health Month, Ubiqi, and Freckle.

Migraine Friendly

The color scheme and screen layout should be easily accessible to users experiencing a migraine at the time that they are using our software.

Fun

The software must be fun for the user to use and not feel tedious each time the user enters a report.

Performance Requirements

Requests to Database Should Take Less than 100ms

Any database queries should be able to be executed in under 100ms.

Other Performance Requirements

Heroku Cloud Application Platform will handle all issues with peak load, concurrency, and other related issues within its operating constraints.

Reliability Requirements

Mean Time Between Failures of 6 Months

Our software should not have unplanned downtime any more often than every six months. Realistically, our software should only experience unplanned downtime when the Heroku cloud is taken down and should be available at all other times

System Should Store Data Given by Users

Our software should be able to store the data given by its users. This may include but not be limited to migraine reports, personal data, and personal settings.

System Should Report When it Fails

Our software should report when it experiences and unexpected failure, this report will be sent by email to a pre programmed email address, preferably belonging to a member of the software's support staff.

Supportability Requirements

Maintainability

Our code must be made in such a way that an outside party can easily maintain it and change it if they are given the proper system privileges.

Easily Expandable

The program will need to be easily expandable to allow new features to be added by the client at a later date.

Help Menus

The application will contain help menus to help answer user questions

Documentation

Readme File

We must maintain a well maintained and up to date Readme file to ensure that any future developers that need to perform maintenance on our system can become familiar with it and prepare their coding environment properly in a fast and easy manner.

Installation

Installation Process

The installation process for our software consists of uploading the software application to the specified Heroku server therefore making it available to the users.

Legal

Terms and Conditions

Our software must contain a terms and conditions confirmation on account creation and must also guarantee the privacy of the user's data within our system. It will ensure that we will never distribute or share private information of any of our users.

Licensing Requirements

All Resources Used Will be Licensed

All resources used by the system and in the development of the system will be either developed by the development team or be owned by the client. This will ensure that we have no legal issues with regard to licenses in the development of the project.

Hardware Software Interface

Our project does not interface with any hardware or software components. It will be hosted using Heroku as listed in the constraints but this does not qualify as an interface as it is simply a host and does not directly interface with our software.

User Interface

See attached prototypes.

Coding Standards

The team will be following the standards as described at <http://bit.ly/tS8y3A>. Listed are some of the key standards listed on the site:

- Use 2 space indent, no tabs.
- Use Unix-style line endings.
- Use spaces around operators, after commas, colons and semicolons, around { and before }.
- No spaces after (, [and before],).
- Use two spaces before statement modifiers (postfix if/unless/while/until/rescue).
- Indent when as deep as case.
- Use an empty line before the return value of a method (unless it only has one line), and an empty line between defs..
- Use empty lines to break up a long method into logical paragraphs.
- Keep lines fewer than 80 characters.
- Use def with parentheses when there are arguments.
- Never use for, unless you exactly know why.
- Never use then.
- Use when x; ... for one-line cases.
- Use &&|| for boolean expressions, and/or for control flow. (Rule of thumb: If you have to use outer parentheses, you are using the wrong operators.)
- Use snake_case for methods.
- Use CamelCase for classes and modules. (Keep acronyms like HTTP, RFC, XML uppercase, unless a Rails model. Support for acronyms is coming soon.)
- Use SCREAMING_SNAKE_CASE for other constants.

Change Control

The client will be providing us with the software to deal with change requests and has asked we not concern ourselves with setting a system up at this time. We do not yet know what software she will be using.

Test Cases

Account Creation

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User Enters Valid Information and does not cancel	Basic Flow	-	-
User Cancels Account Creation	Basic Flow	A	-
User Enters Invalid Information or Leaves Fields Blank	Basic Flow	B	-
User Enters invalid information and cancels operation	Basic Flow	B	A
Database is down and cannot process request	Basic Flow	C	-
Database is Down and User Cancels	Basic Flow	C	A

Test Case ID	Scenario	Additional Information	Expected Results
1	User Enters Valid Information and does not cancel	-	User creates a valid account to be stored on the database
2	User Cancels Account Creation	-	User is returned to homepage and an account is not created
3	User Enters Invalid Information or Leaves Fields Blank	-	User is prompted to re-enter the incorrect information.
4	User Enters invalid information and cancels operation	-	User is returned to homepage and an account is not created
5	Database is down and cannot process request	The validity of the information entered is not a determining factor in this test case.	An error message is displayed and the user is returned to the homepage. An account is not created.
6	Database is Down and User Cancels	-	User is returned to homepage and an account is not created

User Authentication

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User Correctly enters a valid username and password	Basic Flow	–	–
User enters an invalid username and/or password	Basic Flow	A	–

Test Case ID	Scenario	Additional Information	Expected Result
1	User Correctly enters a valid username and password	–	User is logged in and sent to the homepage. User privileges are granted.
2	User enters an invalid username and/or password	The error message displayed is the same regardless of which category is wrong	An error message is displayed and users are asked to correct the wrong information.

Changing Account Preferences

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User Changes their Account Preferences with no Problems	Basic Flow	–	–
User enters invalid information	Basic Flow	B	–
User Cancels change to account preferences	Basic Flow	A	–
User enters Invalid information and the cancels	Basic Flow	B	A

Test Case ID	Scenario	Additional Information	Expected Result
1	User Changes their Account Preferences with no Problems	User validly changes all information that they wished to change	Account preferences are changed. User is returned to Homepage
2	User enters invalid information	–	User is prompted to correct invalid information.
3	User Cancels change to account preferences	–	User is returned to homepage. No Information is changed
4	User enters Invalid information and the cancels	–	User is returned to homepage. No Information is changed

Logout

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User view Analytics with no problems	Basic	–	–
User cancels the operation	Basic	A	–

Test Case ID	Scenario	Additional Information	Expected Result
1	User presses the log out button	–	User is logged out and returned to the homepage
2	User cancels the operation	–	User is returned to the homepage

User is Able to Edit Previous Report

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User correctly submits changed report	Basic Flow	–	–
User incorrectly fills out report change.	Basic Flow	B	–
User cancels report change process	Basic Flow	A	–
User incorrectly Fills out then cancels the report	Basic Flow	B	A

Test Case ID	Scenario	Additional Information	Expected Result
1	User correctly submits changed report		Changed Report is correctly submitted to Database
2	User incorrectly fills out report change.		User is prompted to correct invalid fields
3	User cancels report change process		User is returned to Homepage, changed Report is not submitted
4	User incorrectly Fills out then cancels the report		User is prompted to correct invalid fields but chooses to cancel and is returned to the Homepage. Report change is not submitted.

User is Able to Export to PDF

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User Exports to a PDF	Basic Flow	–	–
User Selects cancels operation	Basic Flow	A	–
User Views PDF in browser	Basic Flow	B	–
User views the PDF in the browser and then cancels	Basic Flow	B	A

Test Case ID	Scenario	Additional Information	Expected Result
1	User Exports to a PDF	–	User downloads a PDF of the migraine data
2	User selects cancels operation		User is returned to home page
3	User Views PDF in browser	–	User views a PDF that is the same as the downloaded one
4	User views the PDF in the browser and then cancels		User views a PDF and then is returned to the homepage

Logout

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User Presses the log out button	Basic	-	-

Test Case ID	Scenario	Additional Information	Expected Result
1	User presses the log out button	-	User is logged out and returned to the homepage

Delete Account

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User deletes the account and then correctly enters their password	Basic Flow	-	-
User cancels deleting the account	Basic Flow	A	-

Test Case ID	Scenario	Additional Information	Expected Result
1	User deletes account with no errors	-	Users account is permanently deleted from the database
2	User cancels deleting the account	-	Users account is not deleted

User Submits Reports

Scenario	Originating Flow	Alternate Flow	Next Alternate Flow
User submits all reports with valid information	Basic Flow	-	-
User cancels operation at any time	Basic Flow	Alternate Flow A	-
User enters invalid information in a report	Basic Flow	Alternate Flow B	-

Test Case ID	Scenario	Additional Information	Expected Result
1	User submits all reports with valid information	-	User successfully submits reports
2	User cancels operation at any time	-	User is brought back to home page
3	User enters invalid information in a report	-	User is brought back to the same page that they just filled out, shown an error message, and allowed to reenter the data.

Usability Report

The following sections detail the process, analysis, findings, and presentation of the usability tests that our project team performed. The findings were used in the revision of our interface design to improve our software's interface. All other sections serve to document the methods by which we carried out our studies.

Process

For our usability tests we started the process by reading a script, (see Appendix A for script), to the current test participant, which outlined what we would need them to do and gave them the background information on the project. After that we had them sign our consent form (Appendix B) and then fill out the pre-test questionnaire (Appendix C). From there we escorted the volunteer to the lab where we gave them a form detailing the tasks (Appendix D) we wished for them to complete with the software and how we wished for them to conduct themselves during the test.

After the current test subject had completed all of the tasks, we would give them a different questionnaire to fill out (Appendix E). When they had finished filling out the questionnaire, their participation in our usability tests would be over. We then moved on to the next subject.

Analysis

With the use of computers, and most of them have high levels of experience with the operations of computers. Most have no current method of tracking migraines, with one user currently using a paper journal. This means that most of the subjects have no experience with tracking migraines, and that if they find our system difficult to use then it will be even more difficult for our general market.

However, on a scale of one to five, with one being very difficult to use and five being highly intuitive to use, all of the users rated our system as at least a four, with at least one rating it to be a five. From this we can conclude that our system is intuitive, at least for people with some experience with computers. Most of them said that our color scheme would not aggravate their migraine, if they operated the system with one. For the participants who felt like they would use any sort of migraine tracking system, all of them said that they would consider using this system. They also made suggestions, most of which were along the lines of "More information on the Export Data page" or "The logout option should be part of the task bar". We used those as a basis for our usability findings.

Findings

The following sections contain the findings of our usability study and recommendations for changes to be made to our interface design.

Buttons on Homepage

Many of the participants mentioned, either during testing or on the post testing questionnaire, that having a large buttons on the Home page that takes users straight to key features would be nice
Recommendations: Add buttons onto the homepage.

Change How Submit Works

Users mentioned that the submit reports sequence was confusing and unintuitive.
Recommendations: Alter the submit sequence so that they submit reports one at a time, not in a row.

Export

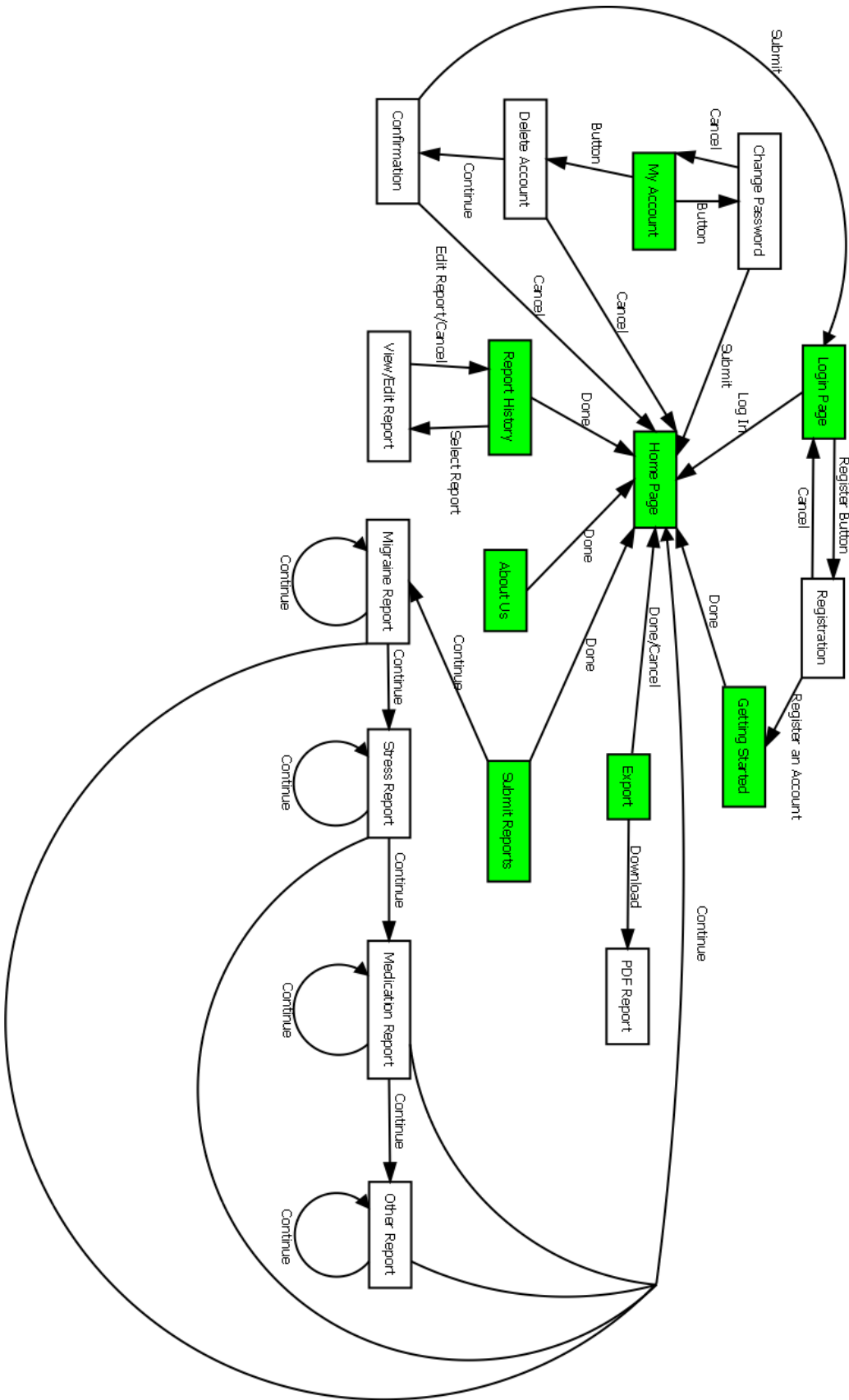
Users said that the export page was vague, and offered no explanation as to what they should do.
Recommendations: Add help to the main site, and add a description to the export page.

Log out

The log out button was not part of the top menu bar, and several users noted this and mentioned it as something to fix.
Recommendations: Append the menu bar to have the log out button

Interaction Architecture

The following page contains our interaction architecture. All of our web pages with the exception of our log in page contain a button bar on the top of the page with links to key pages of our website. Because of this, these key pages are accessible at all times while using our application (except for the login page). To conserve space in our diagram we have omitted any connections using the top button bar. All colored boxes in the diagram can be accessed from all pages (except login page) using the top button bar.



Initial Interface Design

The interface design used during usability testing can be found on the usability testing computer in F225B or can be provided upon request.

Revised Interface Design

The revised interface design can be found on the usability testing computer in F225B or can be provided upon request.

Glossary

Heroku

A cloud service that acts as the platform that our application will be running on. It handles software scaling and other performance constraints.

Cucumber

A behavior driven testing framework for Ruby programming language.

Ruby

A dynamic, reflective, general-purpose object-oriented programming language that combines syntax inspired by Perl with Smalltalk-like features

Rails

An open source web application framework for Ruby programming language

GitHub

A web based hosting service for software development projects using Git revision control system

Git

A distributed revision control system with an emphasis on speed

Revision Control

Software that tracks many developers work on a given project without needing to be connected to a common network

Web Application

An application that is accessible over a network such as the internet or an intranet.

Readme

Contains information about other files in a directory or archive. Often distributed with computer software.

References

Skype meeting with Eliza Brock on 09/11/2011

Skype meeting with Eliza Brock on 9/28/2011

Skype meeting with Eliza Brock on 10/10/2011

Skype meeting with Eliza Brock on 10/19/2011

Ubiqihealth.com

<http://pub.cozmixng.org/~the-rwiki/rw-cgi.rb?cmd=view;name=RubyCodingConvention>

- "RubyCodingConvention"

En.wikipedia.org

Change Log

Version	Date	Changes
3.1	10/26/2011	Change Tracking Started
3.2	10/26/2011	Consolidated the Activity Report Use Case and the Previous Migraine Use Case into one unified use case process and performed strikethroughs on the related sections in Use Cases, Feature Mapping, and Test Plan sections. Added TO DO section to the Test Plan section for the new use case. Still need alternate flows for the new use case. AFFECTED SECTIONS: Use Cases, Feature Mapping, Test Plan
4.1	10/27/2011	Revising the current Milestone. Editing Coding Standards, Change Controls, and Test Cases
4.2	10/29/2011	Milestone 4 completed and submitted
5.1	11/8/2011	Added necessary sections for milestone 5 and updated the executive summary and introduction to the new document format. Also removes storyboards
6.1	11/10/2011	Removed strikethrough information from previous versions
6.2	11/10/2011	Compiled corrections to the document.
6.3	11/11/2011	Replaced DFD's, final version