

Software Requirements Specification

for

Hot Logbook

Version 0.1

Prepared by

Group Name: *Sasswords Puck*

Jesse Griffin
Patrick Kent
Pierson Cavulli

11641870
11019238
11608370

jesse.w.griffin@wsu.edu
patrick.kent@wsu.edu
pierson.cavulli@wsu.edu

Date: 7 October 2018

Contents

REVISIONS	III
1 INTRODUCTION.....	1
1.1 DOCUMENT PURPOSE.....	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	1
1.5 DOCUMENT CONVENTIONS.....	1
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
2 OVERALL DESCRIPTION.....	3
2.1 PRODUCT PERSPECTIVE.....	3
2.2 PRODUCT FUNCTIONALITY.....	3
2.3 USERS AND CHARACTERISTICS.....	4
2.4 OPERATING ENVIRONMENT	4
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	4
2.6 USER DOCUMENTATION	4
2.7 ASSUMPTIONS AND DEPENDENCIES.....	4
3 SPECIFIC REQUIREMENTS.....	5
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	5
3.2 FUNCTIONAL REQUIREMENTS.....	6
3.3 BEHAVIOUR REQUIREMENTS	6
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	7
4.1 PERFORMANCE REQUIREMENTS	7
4.2 SAFETY AND SECURITY REQUIREMENTS	7
4.3 SOFTWARE QUALITY ATTRIBUTES	7
5 OTHER REQUIREMENTS	8
APPENDIX A – DATA DICTIONARY	9
APPENDIX B - GROUP LOG	10

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
V0.1	Patrick Kent Jesse Griffin Pierson Cavulli	Initial revision. Including project scope and initial figures and descriptions of processes.	10/7/18

1 Introduction

1.1 Document Purpose

This Software Requirements Specification (SRS) document describes the inner and outer workings of the Hot Logbook v0.1. It will describe the scope, identify the audience, define project-specific acronyms and language. Also outlined in this document will be an overall description, including basic functionality, functional and non-functional requirements, as well as any other software requirements.

1.2 Product Scope

Hot Logbook is a web application that will act as a repository and generator for user passwords for various software services and websites. It will take into account the password requirements of the service or website and store these in a local encrypted database. The goal of Hot Logbook is to provide users with strong passwords to use for all their services, as well as an encrypted storage platform to save them all.

1.3 Intended Audience and Document Overview

This document is intended to be used and read by clients who will use this web application, as well as the professor who will evaluate its software worthiness.

1.4 Definitions, Acronyms and Abbreviations

App: Application
CSS: Cascading Style Sheets
CSRF: Cross-Site Request Forger
DB: Database
FAQ: Frequently Asked Questions
HTML: Hypertext Markup Language
JS: JavaScript
RFI/LFI: Remote/Local File Inclusion
SQL: Structured Query Language
SRS: Software Requirements Specifications
UI: User Interface
XSS: Cross-Site Scripting

1.5 Document Conventions

This document follows IEEE formatting unless otherwise specified.

1.5.1 Naming Conventions

Naming conventions with respect to this document and this product will be *generally* descriptive, as opposed to specifically descriptive (e.g. `print()` versus `printOneLineOnly()`). This is to avoid

assigning malleable or changing product or function-specific requirements to non-changeable portions of the product itself, such as function names or prototypes.

1.6 References and Acknowledgments

Not applicable for v0.1.

2 Overall Description

2.1 Product Perspective

This is a new self contained web application. The Hot Logbook provides the user with the following. A login screen, a user profile, new user page, generate password page, and Custom service page. The login page will be the landing page (first page that the user views). The user profile will be displayed upon a successful login. The user profile page will allow the user to manage the passwords that they have for different accounts; as well as get to the other pages. Where the user will be able to generate new passwords (the generate page), and set password requirements for certain websites, such as cannot contain a special character, or password must be between eight and fourteen characters.

2.2 Product Functionality

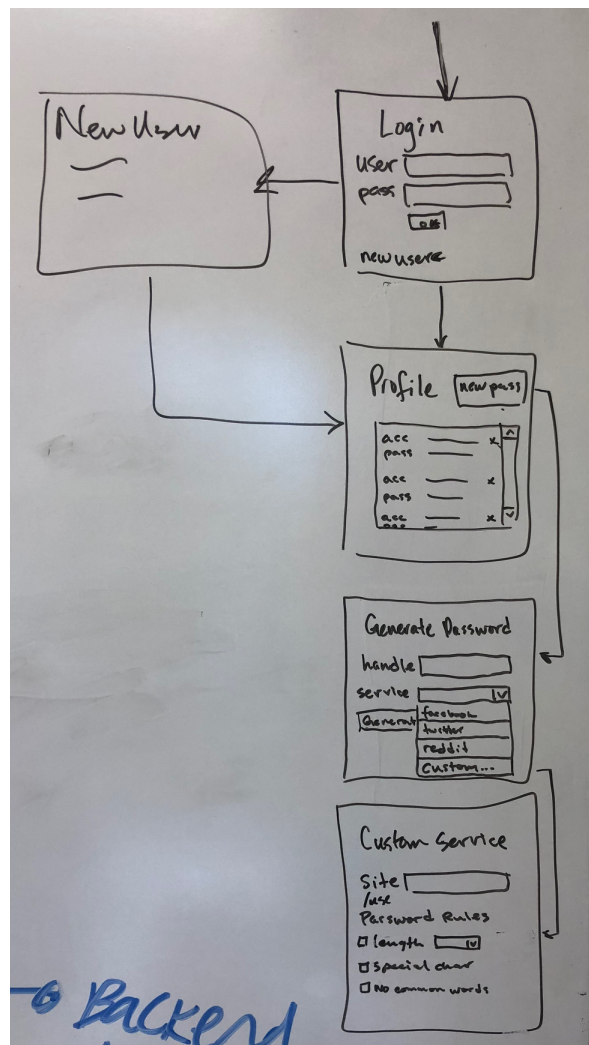


Figure 1: Hot Logbook Data Flow Diagram

- Generate secure passwords for an account specified by the user
- Logs all usernames and passwords to a database
- Allows for new accounts to be added to the users profile
- Allows the end user to specify password rules that they need.

2.3 Users and Characteristics

Anyone who wants to remember their password, or would like to generate a secure password. The most important users for this product are business men/women who need secure passwords for their organization.

2.4 Operating Environment

This project is a web application. Therefore it should run on any modern device capable of using Google Chrome or another suitable Browser. Hot Log book will work on Linux, Windows and Mac because it is a web application. This web application will be able to peacefully coexist with IntelliJ.

2.5 Design and Implementation Constraints

Some constraints that could effect the final project are security, and the use of different programming languages. Security is an issue with any project, specifically web applications, and since we are new to Javascript we do not yet know of vulnerabilities that exist within the language. Building a database will also be difficult. Given that there will be a lot of passwords that need to be logged we were thinking of using some kind of structured query language to manage the data. This opens up a whole new can of worms because we will need to learn how to create an SQL database and defend it against common kinds of database attacks, such as SQL injection.

2.6 User Documentation

This software will include an FAQ section, which will be hosted on the web page. The FAQ section will display the answers to frequently asked end user questions. This page will also include a short description of how to successfully interact with the website.

2.7 Assumptions and Dependencies

Some assumptions that I am making are: each user will enter correct password requirements for whatever website that the password is needed for. The user will only enter nice input IE nothing intended to cause the web page to behave in an inappropriate manner. Lastly, the user will not login from more than one location at a time.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The first screen the user will see will be a login screen. This will have a box for inputting their username and password for our service. There will also be a link for the user to create an account with us which will lead to our next page.

The second screen will be for creating an account. All this page will do is ask the user to create a user name for our website and password. After submitting this information, the user will be directed back to the first screen to log in.

The third page will be the home page for the user. It will show them all the accounts and passwords they have created previously, or will be blank if they are a new user. From this page, the user will be able to delete previously saved accounts or will use the new password link to save a new username and generate a password.

Clicking the new password button will send the user to the fourth page. This page will ask for the users username for whatever account they are trying to generate a password for. They will also use a drop box to confirm what site they are generating a password for. After clicking submit, the user will be brought back to the home page where their new account and generated password will be listed with the others,

Finally, if the user selects custom from the drop box on the generate password page, they will be lead to our fifth and final page. This page will ask questions about what is expected from the website in regards to password expectations. This will allow the website to generate a password for the user that meets the requirements of the site.

3.1.2 Hardware Interfaces

The hardware interfaces this program will use will be the users computer and the server that both communicates to the user and stores the database of users and passwords.

3.1.3 Software Interfaces

This program will be able to run on all OS that can run Google Chrome. JavaScrips must also be allowed so that the logic of the program can be implimented.

3.1.4 Communications Interfaces

This site will be run through a web browser (Google Chrome.) Due to the sensitive imformation begin recorded, the comunication standard that will be implemented with be HTTPS. The data will need to be encrypted to protect the users information and that will be done through MongoDB.

3.2 Functional Requirements

1. Access database to confirm the username and password of a user who is already in the database.
2. Write to the database for both new users and new account information that is provided by an existing user.
3. Generate a password through a pre-existing algorithm
4. Generate a password with the specifications taken from the user
5. Connect graphical interfaces in flow designated in 3.1.1.

3.3 Behaviour Requirements

3.3.1 Use Case View

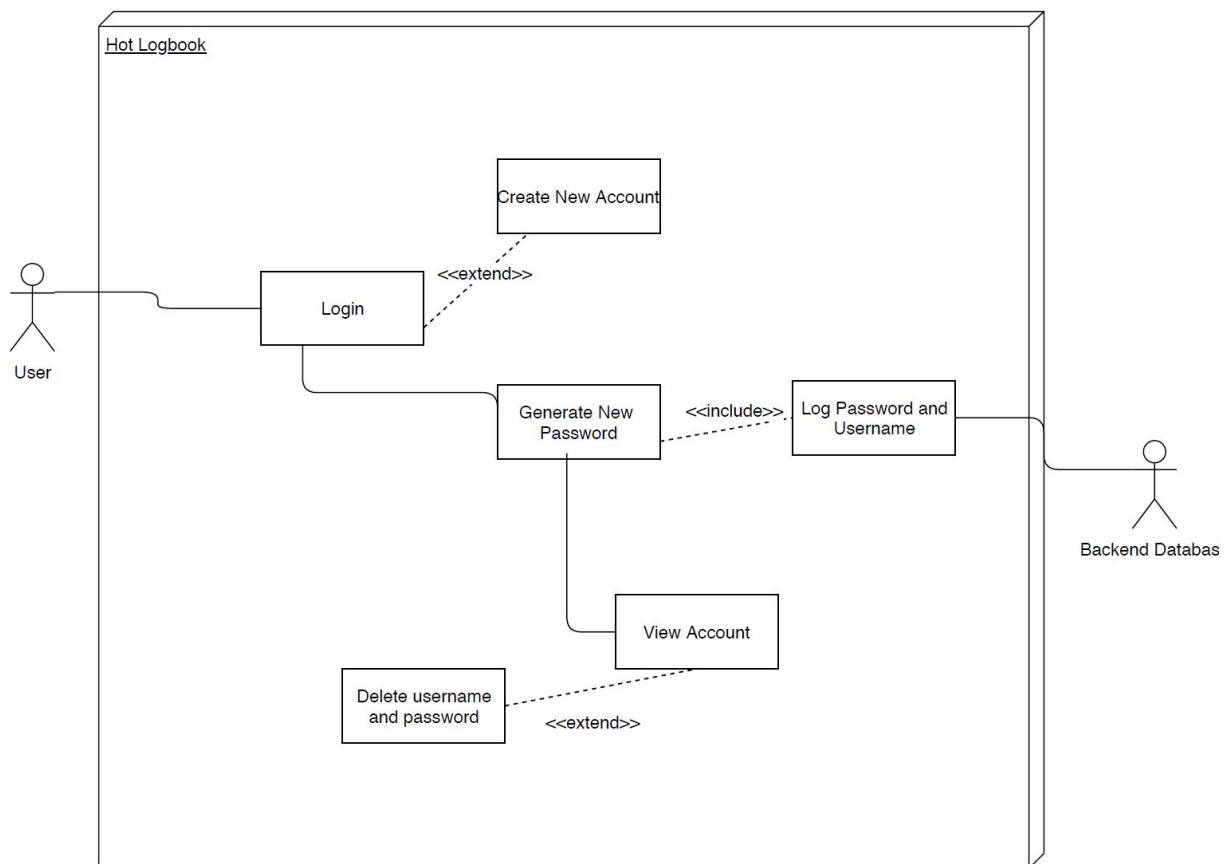


Figure 2: Use case diagram of Hot Logbook.

4 Other Non-functional Requirements

4.1 Performance Requirements

1. Generating a password should not take more than 10 seconds.
2. All transitions between pages should take less than 5 seconds.
3. All operations should be ran at under 10 seconds and if they are not, users internet connection should be tested.

4.2 Safety and Security Requirements

Safety requirements are as follows: This product shall not physical harm to the user or reflect the clients organization in an inappropriate way.

The clients that will be using this Web-App will be expecting their usernames and passwords to be kept secure. It is for this reason that we are providing the following security requirements.

- Defend against XSS (considering)
- Protect the web application from remote file inclusion (RFI) and local file (LFI) inclusion attacks (considering)
- Defend the contents of the SQL database by sanitizing user input. (No SQL injection!)
- Protect the client from Cross-Site Request Forgery attacks (CSRF). (considering)
- Command injection (considering).

4.3 Software Quality Attributes

4.3.1 Usability

The system must be highly usable by any standard user seeking to generate passwords for any service or website. Each page will have text in a web-appropriate font, sized to be readable and legible for anyone accessing it. The number of buttons per page to perform actions will be kept to a minimum.

Ideal minimum browsers to support: Google Chrome, Mozilla Firefox, and Microsoft Edge.

Minimum OSes to support: Windows 10, Apple Mac OS 10.1x.

4.3.2 Reliability

Hot Logbook will rely on it's own back-end database, reducing the need for a third-party system. It will not work when the internet is down, but ideally having the user store their own local database will reduce security risk.

4.3.3 Maintainability

Sasswords Puck will seek to make the code for Hot Logbook modular, allowing for flexibility when making major changes. Modularity will also make Hot Logbook scalable, allowing for implementation of future functionality.

5 Other Requirements

This section will be utilized on an as-needed basis.

Database requirements:

- Must be scalable.
- Must be able to manipulate using JS.
- Must be secure.

Appendix A – Data Dictionary

This section will be filled out once the first prototype has been completed.

Appendix B - Group Log

GitHub commits can be viewed here: <https://github.com/easypat/CS320Project>

Group meetings and their minutes will be included as attachments following this page.

Group Meeting #1 – 10 October 2018 3pm-5:30pm

1. Attendance

- Jesse Griffin
- Patrick Kent
- Pierson Carulli

2. Location

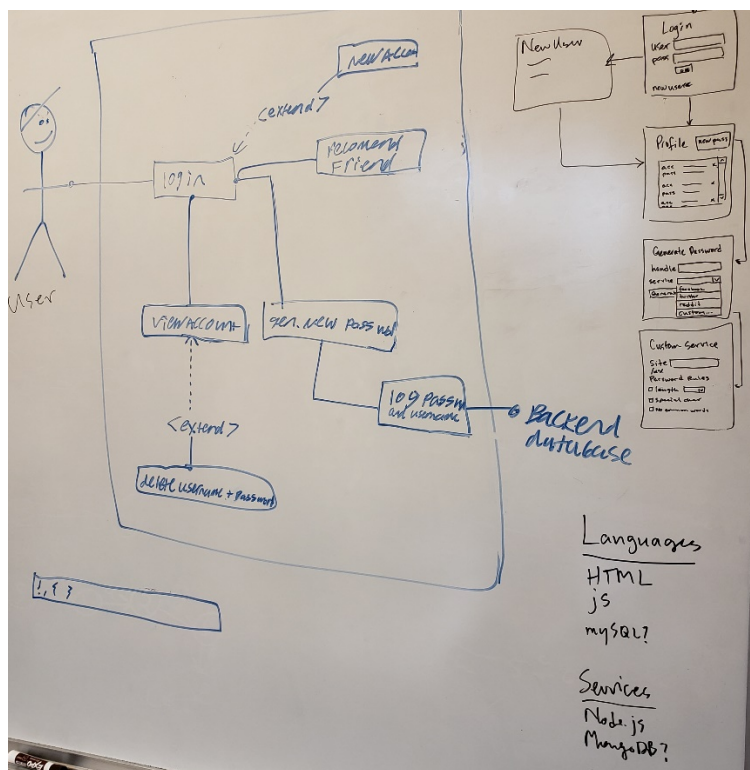
- WSUV Library

3. Discussion

- Come up with use case diagram for our user
- Define languages used for our project
- Draw out basic design for our 5 pages that will be navigated by user
- Discuss security concerns for project
- Figure out git and git conflicts before end of meeting
- Established workflow to avoid merging conflicts

4. Post Meeting Consensus

White board notes:



How we broke up SRS:

Jesse – Section 3, Section 4.1

Patrick – Section 1, Section 4.3

Pierson – Section 2, Section 4.2