Software Requirements Specification

for

CodeIt

Version 1.0 approved

Prepared by Stanley Vossler, Travis Thomspson, Daniel Clemente, and Jacob Scarf

using teamname std;

7 December 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Revision	12/7/2023	Initial version of the SRS document.	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to describe the website "CodeIt". It will explain the purpose of the website, interface of the website, functionality available to users of the website, operating environment the website runs on, constraints of the database, and assumptions about website security and functionality.

1.2 Document Conventions

"website": the CodeIt website

"users": anyone visiting and interacting with the website

1.3 Intended Audience and Reading Suggestions

Developers: are in charge of creating the platform, should begin by studying the project overview to comprehend the broad objectives. They should next concentrate on the in-depth explanations of the system's functionalities, content curation and moderation algorithms, collaboration tools, and system architecture. They should also carefully read the sections on data security, scalability, and privacy.

Marketing Staff: the sections on the issue description and project overview will shed light on "CodeIt's" distinctive selling points. Formulating the marketing plan will be aided by knowing the objectives and anticipated results. Additionally, as they might be important components in advertising materials, they should be aware of user reputation and trust systems.

Users: they should focus on the sections detailing the external interface requirements as well as the performance requirements. They will gain an understanding of the platform's features, how to use it, and tech specifications needed from these sections.

Testers: to gain the platform's intended operation, they should concentrate on the characteristics of the system and the anticipated results. Planning suitable load and stress testing also depends on the performance section.

Documentation Writers: they should read the paper in its entirety since they require a thorough understanding of the complete system in its entirety.There should be more attention given to the overall description and the technical specifications, since end users will require concise and understandable explanations for these.

The goal of the text is to first provide all readers with a general overview of the project, then move on to more in-depth and technical sections. It is recommended that each reader begin with the introduction in order to grasp the background, and then move on to the sections that are most pertinent to their particular role in the project.

1.4 Product Scope

"Codeit," a forum website designed to promote knowledgeable, productive, and cooperative online conversations, especially with an emphasis on coding and programming subjects, is the software that is being specified. The goal of "Codeit" is to tackle issues that are frequently encountered on online discussion boards, including toxicity, echo chambers, information overload, false information, and collaborative constraints. The principal advantages of "Codeit" are the cultivation of a civil and knowledgeable community, the encouragement of the exchange of varied perspectives, and the facilitation of cooperative knowledge generation and issue resolution.

1.5 References

www.codeit.com

2. Overall Description

2.1 Product Perspective

The website will be accessed through widely available internet browsers on any operating system. User and content data will be stored in-house. The only data users will have access to are comments and posts made by other users. The website essentially is just the interface between users and their content.

2.2 Product Functions

New users should be able to register for an account.

Users should be able to navigate around the website to read posts and comments.

Users should be able to make posts and comments.

2.3 User Classes and Characteristics

There is only one class in the current version of the website, and that is the "user". Every user has the same security level and access to functionality. Aside from basic navigation of the website, functionality available to the user will be viewing and making posts and comments. The user base is expected to be tech savvy, given our intended audience is programmers.

2.4 Operating Environment

User environment:

operating systems: any
browsers: any
device: desktop and mobile

Development environment:

database: SQLite
backend: Python, Flask
frontend: CSS, HTML, JavaScript, Jinja2

Client environment:

server management is client's choice
database storage is client's choice

2.5 Design and Implementation Constraints

Database growth and scaling limits how many users can be served.

Speed of access for users will depend on availability and proximity of servers.

The website is only available in english.

The client will be responsible for maintaining the database. Since the website just displays information in the database, the quality of the website will depend upon well kept data.

2.6 User Documentation

A manual will be provided to the customer detailing the database structure and user functions available that can manipulate the database. Since the client will be responsible for this, we won't need to give specific instructions for maintenance; rather, the layout for guidance.

2.7 Assumptions and Dependencies

The client will properly maintain records in the database.

Database, frontend, and backend web frameworks will be stable over time.

Security techniques employed will keep the website functioning properly. Consistent maintenance and security updates will be made to keep normal functionality.

3. External Interface Requirements

3.1 User Interfaces

A standard QWERTY or DVORAK keyboard and mouse is sufficient to interact with the website. HTML5 input forms take input directly from the keyboard. Users can also access the website via mobile phone (Android or iPhone), tablet (iPad, Samsung Tab, Kindle Fire), desktop or laptop.

3.2 Hardware Interfaces

A USB or Bluetooth keyboard, mouse and monitor is sufficient to use the website. 67 MB of RAM is the absolute minimum required to host the website. An HDD or SSD with standard buses can be used to host it and a monitor with VGA or HDMI connectors can be used to view the interface.

3.3 Software Interfaces

The only software needed to run your own instance of the app is Python 3, Flask, and the SQLite3 Python module, as well as a standard, modern web browser and any modern operating system like MacOS, Windows 7, 8, 10, or 11, and any Linux or BSD distribution that receives regular security updates.

3.4 Communications Interfaces

A standard, modern website that supports the TCP/IP, IPv4 and HTTP protocol. The Gopher and Gemini protocols are not supported.

4. System Features

4.1 Home Page

4.1.1 Description and Priority

The home page is where posts are dynamically passed from the database, depending on what subforum is currently selected. Posts that are displayed can be clicked on and sent to the post page for its contents to be displayed. High priority.

4.1.2 Stimulus/Response Sequences

Stimulus 1: The user types in the url of the website and loads onto the page.

Response 1: Backend calls index function and client connects to the ForumPosts database and selects all of the posts that belong to the currently selected subforum.

4.1.3 Functional Requirements

REQ-1: The user has an internet connection.

REQ-2: The user has a web browser installed on their machine.

4.2 Post Page

4.2.1 Description and Priority

The post page is where the post's content is displayed and also whatever comments that are made under the post. High priority.

4.2.2 Stimulus/Response Sequences

Stimulus 1: The user clicks on the post title link on a post listing from the home page.

Response 1: The PostId is passed to the backend which then the post content to the post page.

Stimulus 2: The user clicks on the post title link on a post listing from the profile page.

Response 2: The PostId is passed to the backend which then the post content to the post page.

4.2.3 Functional Requirements

N/A

4.3 Profile Page

4.3.1 Description and Priority

The profile page is similar to the home page, but only displays the posts of the profile the user is visiting or displays the logged in user's posts if they select the "Profile" link. Medium priority.

4.3.2 Stimulus/Response Sequences

Stimulus 1: The user clicks on the username preceded by "Posted by:" on a post listing.

Response 1: The UserId is passed back to the backend where it then sends all of the posts that were posted by that user.

Stimulus 2: The user clicks on the "Profile" link displayed in the top navigation bar

Response 2: The UserId of the logged-in user is passed to the backend where it then sends all of the posts that were posted by the logged in user.

Stimulus 3: The user clicks on the username that appears above every comment in the comment section.

Response 3: The UserId is passed back to the backend where it then sends all of the posts that were posted by that user.

4.3.3 Functional Requirements

REQ-1: In order to view the "Profile" link on the navigation bar, the user must be logged-in

4.4 Subforum Navigation

4.4.1 Description and Priority

On the left of the navigation bar reads "CodeIt/[insert subforum name]". The text shown as the subforum name (All, Python, JavaScript, C++, or Java) is an HTML select tag, and on select change triggers the home page to re-render, displaying only the posts of the newly selected subforum. Low priority.

4.4.2 Stimulus/Response Sequences

Stimulus 1: The user clicks on "CodeIt/All"

Response 1: A dropdown menu appears showing various subforums the user can pick from.

Stimulus 2: The user selects a subforum from the dropdown menu.

Response 2: This triggers the on select change function to fetch all the posts that belong to the subforum and passes them to the reloaded home page.

4.4.3 Functional Requirements

N/A

4.5 Posting

4.5.1 Description and Priority

User fills out a form and must provide inputs for the following fields: Title, Post Content, Subforum which is then inserted to the ForumPosts database. High priority.

4.5.2 Stimulus/Response Sequences

Stimulus 1: The user tries to create a post without a title.

Response 1: The backend sends a flash message that says “Your post needs a title and content.”

Stimulus 2: The user tries to create a post without content.

Response 2: The backend sends a flash message that says “You post needs a title and content.”

Stimulus 3: The user fills all the fields and presses the “Create Post” button.

Response 3: The backend connects to the ForumPosts database and then inserts the new post into the database, closing the connection after.

4.5.3 Functional Requirements

REQ-1: The user must be logged in in order to have the ability to create a post.

4.6 Commenting

4.6.1 Description and Priority

The logged in user can post non-empty comments on any post. Medium priority.

4.6.2 Stimulus/Response Sequences

Stimulus 1: The user clicks on a post from the home page.

Response 1: The PostId is passed back to the backend where it then sends all of that posts information to be displayed in post.html

Stimulus 2: The user tries to comment while not logged in

Response 2: The backend sends a flash message that says “You must be logged in to comment!”

Stimulus 3: The user tries to submit an empty comment

Response 3: The backend sends a flash message that says “You can’t post an empty comment!”

4.6.3 Functional Requirements

REQ-1: The user has to be logged in order to comment.

REQ-2: The comment must not be empty.

4.7 Upvoting/Downvoting

4.7.1 Description and Priority

On every post listing, there is an orange upvote button and an orange-red downvote button. The logged in user can click on either to increment or decrement the current vote total. Low priority.

4.7.2 Stimulus/Response Sequences

Stimulus 1: The user tries to click on the upvote button when not logged in.

Response 1: The backend sends a flash message that says “You need to be logged-in to upvote.”

Stimulus 2: The user tries to click on the downvote button when not logged in.

Response 2: The backend sends a flash message that says “You need to be logged-in to downvote.”

Stimulus 3: The logged-in user clicks on the upvote button.

Response 3: The backend is passed the PostId of the post that was upvoted and then updates that row of the PostForum database with the incremented value.

Stimulus 4: The logged-in user clicks on the downvote button.

Response 4: The backend is passed the PostId of the post that was downvoted and then updates that row of the PostForum database with the decremented value.

4.7.3 Functional Requirements

N/A

4.8 Search Bar

4.8.1 Description and Priority

Located in the navigation bar. Typing in queries searches for matches in post title and post content. Low priority.

4.8.2 Stimulus/Response Sequences

Stimulus 1: The user types in a query and presses on the magnifying glass icon or presses enter to execute the search.

Response 1: The backend finds all posts that match the query and passes them to the reloaded home page.

4.8.3 Functional Requirements

N/A

4.9 User Login

4.9.1 Description and Priority

The user fills out the form with their credentials and sends the user back to the home page after a successful login attempt. High priority.

4.9.2 Stimulus/Response Sequences

Stimulus 1: The user types in an incorrect username and tries to login.

Response 1: The backend sends a flash message that says “Invalid username or password.”

Stimulus 2: The user types in an incorrect password and tries to login.

Response 2: The backend sends a flash message that says “Invalid username or password.”

Stimulus 3: The user types in a correct username and password and logs in.

Response 3: The logged-in session variable is set to True and the name session variable is set to the username. The user is then relocated to the home page.

4.9.3 Functional Requirements

REQ-1 The user’s credentials must be in the database in order to login.

4.10 User Registration

4.10.1 Description and Priority

User fills out a form and must provide inputs for the following fields: Email Address, Username, and Password. High priority.

4.10.2 Stimulus/Response Sequences

Stimulus 1: The user tries registering a username that already exists in the database.

Response 1: The backend sends a flash message that says “.”Username taken, please choose another.”

Stimulus 2: The user tries pressing the “Register” button with an empty field.

Response 2: “Please fill out this field.” displays below the input area.

4.10.3 Functional Requirements

REQ-1: The username the user selects must not already be a registered username.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

It is recommendable to host the app on a multicore server if a large user base (e.g. over a million) is expected. Horizontal scaling is recommended. Users are also recommended to access the web interface on multicore and 1GB RAM machines if they’re going to be reading a large amount of posts and comments daily.

5.2 Safety Requirements

Use adequate cooling so the server does not overheat during peak times. Liquid cooling is preferable if heat becomes a big issue. Users should make sure to not expose personal information in any posts or comments and should limit access to minors even though they are allowed to use the app because there is no age requirement.

5.3 Security Requirements

User data will be stored as one-way encrypted hashes. User inputs will be defended against SQL injection attacks.

5.4 Software Quality Attributes

Availability: the website should run constantly with no downtime for maintenance.

Correctness: data should be displayed accurately.

Adaptability: new features should not be difficult to implement.

Usability: the website should be intuitive to use with a preference for ease of reading and writing over ancillary features.

5.5 Business Rules

It is the client's responsibility to moderate content on the website and implement standards of conduct. The developer will simply provide the means for users to interact with the website.

6. Other Requirements

Use of adblocking software by the user may prevent the website from displaying properly.

Appendix A: Glossary

RAM: Random Access Memory
HDD: Hard Disk Drive
SSD: Solid State Drive
HDMI: High-Definition Multimedia Interface
VGA: Video Graphics Array