Software Requirements Specification

for

CodeCraft

Version 1.0

Prepared by

Syed Hussain Ahmed	**CS-21092**	**ahmed4400120@cloud.neduet.edu.pk**
Sameer	**CS-21088**	**sameer4406975@cloud.neduet.edu.pk**
Syeda Aiman Naveed	**CS-21077**	**naveed4410673@cloud.neduet.edu.p**
Malik M. Farooq Ahsan	**CS-21095**	**ahsan4403852@cloud.neduet.edu.pk**

Instructor: Mr Kashif Asrar

Date: 6 December 2023

Contents

Chapter 1

Introduction

This chapter serves as a gateway to understanding the essence of the project. CodeCraft welcomes developers to a new world of coding. Developers no longer need to worry about the hassles of installing compilers, packages, and configuring their system. Here, they can build projects on the go. If they want to challenge themselves or if they want to brush up on their skills learned, they could try out our coding challenges. In the following sections, readers will find a concise introduction outlining the fundamental objectives and significance of the undertaking. Additionally, a brief overview of the content within this chapter will guide readers through key aspects such as the project's background, purpose, and the scope outlined in subsequent sections.

1.1 Document Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the development of a web-based coding platform, CodeCraft. It allows users to create, save, and edit their coding projects while also providing a feature for testing their coding skills through challenges. The document aims to provide a comprehensive understanding of the scope, functionality, and constraints of the proposed system.

1.2 Product Scope

The software to be developed provides an isolated coding environment, akin to Overleaf but for programming languages instead of LaTeX.It also provides an environment similar to HackerRank for solving challenges. The platform facilitates project creation, editing, and storage, encouraging users to

test and improve their coding skills. The benefits include a user-friendly interface, language-agnostic support, and a diverse set of challenges for skill enhancement.

1.3 Intended Audience and Document Overview

This document is intended for developers, potential users, and any stakeholders involved in the development process. It provides an overview of the software requirements, system architecture, and user interactions. Readers are guided through a step-by-step process to understand the project.

- Chapter 1 is the introduction to the product. It discusses the scope,purpose,intended audience,definitions, acronyms, and abbreviations.
- Chapter 2 provides an overview and description of the software.
- Chapter 3 provides the specific requirements the product is expected to deliver. It also provides user interface details that the customers will interact with.
- Chapter 4 specifies the non-functional requirements for the product. Performance, safety, and security requirements are mentioned here. In addition, software quality attributes have been discussed.

1.4 Definitions, Acronyms and Abbreviations

- **API:** An application programming interface is a way for two or more computer programs to communicate with each other
- **Bootstrap:** Bootstrap is a framework to help you design websites faster and easier.
- **CodeMirror:** CodeMirror is a JavaScript component that provides a code editor in the browser. It has a rich programming API and a focus on extensibility.
- **CSS:** Cascading Style Sheets describe how HTML elements are to be displayed on the screen.
- **Figma:** Figma is a collaborative web application for interface design, with additional offline features enabled by desktop applications for macOS and Windows.

- **HackerRank:** HackerRank is the market-leading coding test and interview solution for hiring developers.
- **HTML:** Hypertext Markup Language is the code that is used to structure a web page and its content.
- **React:** React (also known as React.js or ReactJS) is an open-source, front-end, JavaScript library for building user interfaces or UI components.
- **SQLite3:** SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine.
- **VSCode:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

1.5 Document Conventions

In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

1.6 References and Acknowledgments

- J. Doe, Recommended practice for software requirements specifications (ieee). IEEE, New York, 2011.
- L. B. Vikås, "Integrating a web-based editor in the cloud with tdt4100s course wiki," Master's thesis, NTNU, 2015.

Chapter 2

Overall Description

2.1 Product Perspective

The platform is a standalone product designed to provide a collaborative and skill-testing environment. It operates independently but may integrate with external systems for user authentication and project storage. The system interacts with users, a database for project storage, and external challenge modules. The diagram below illustrates the product's high-level components and their interactions.

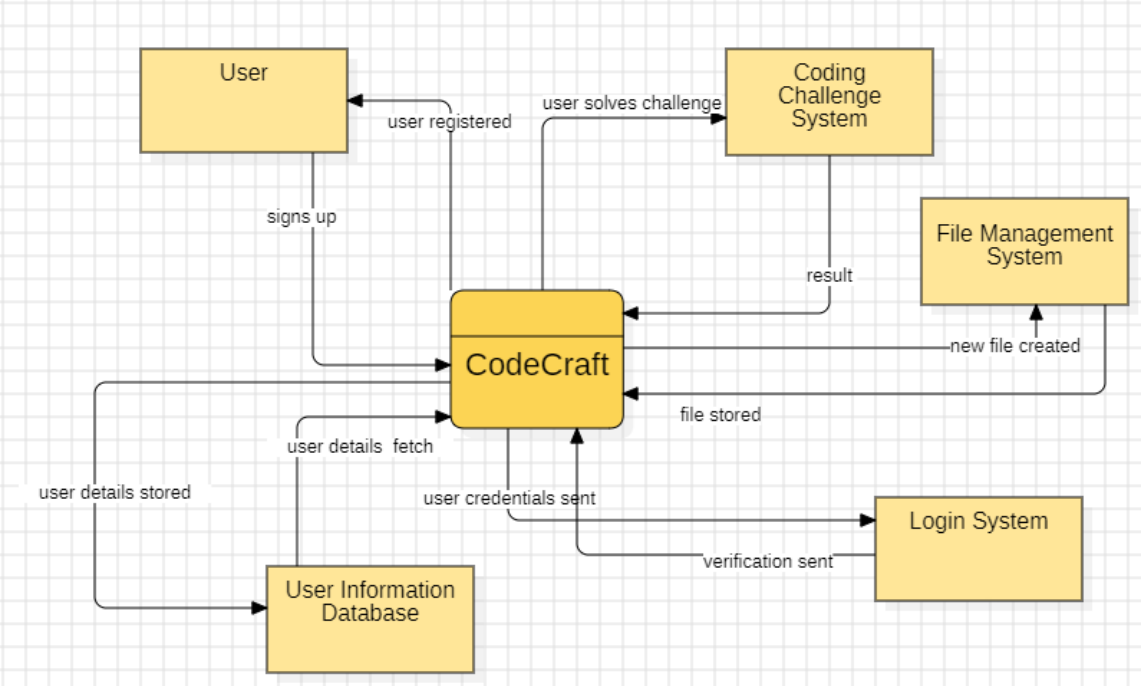

Figure 2.1: Context Diagram

2.2 Product Functionality

The major and main functions that can be done using CodeCraft are listed below. Also, a DFD (data flow diagram) is present below. The system provides

- **User Authentication:** Enable users to create accounts and log in securely.
- **Project Management:** Allow users to create, edit, and save coding projects.
- **Coding Environment:** Offer a user-friendly code editor with syntax highlighting and auto-completion.
- **Challenge System:** Present coding challenges to users for skill assessment.

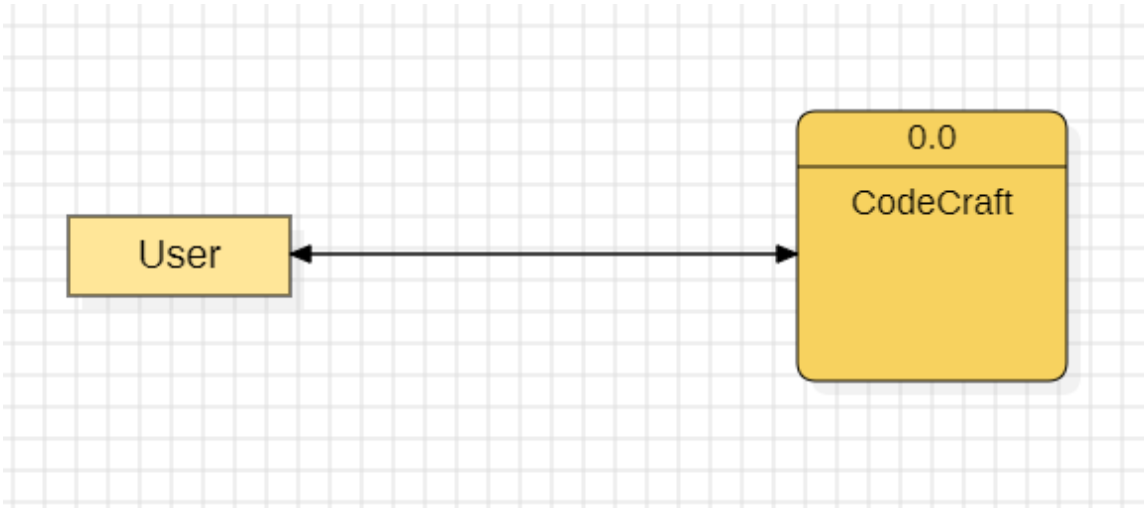

Figure 2.2: Level 0 Data FLow Diagram

2.3 Users and Characteristics

- **Regular Users:** Engage in coding projects and challenges.
- **New Programmers:** Start their programming journey without the hassle of setting up the language environment in their local system.
- **Challenge Solvers:** Focus on improving coding skills through challenges.

The most critical users are the new programmers, as they constitute the primary user base for project creation.

2.4 Operating Environment

The platform will operate in a web-based environment, compatible with major browsers (Chrome, Firefox, and Safari). Minimum platform requirements include a stable internet connection and a modern browser version.

2.5 Design and Implementation Constraints

1. **Language Agnosticism:** The system must support various programming languages.

2. **Security:** Adherence to industry-standard security practices is mandatory.

3. **Browser Compatibility:** The platform should be compatible with common browsers.

4. **Scalability:** The system must handle a growing user base and project repository.

5. **Code Execution:** The execution of user code must be secure and isolated.

These constraints guide the development process, ensuring the system's reliability, security, and performance.

2.6 User Documentation

The user documentation for the coding platform will include the following components:

- **User Manuals:** Comprehensive guides explain how to use the platform, create projects, collaborate, and participate in challenges.

- **On-line Help:** In-context help is accessible within the platform to guide users through specific features, troubleshoot issues, and provide quick references.

- **Tutorials:** Step-by-step tutorials demonstrating key functionalities, such as project creation, collaboration, and challenge-solving, are aimed at helping users become proficient with the platform.

Delivery formats will adhere to industry standards, with user manuals provided in PDF format, online help integrated into the platform, and tutorials accessible through the platform's website.

2.7 Assumptions and Dependencies

Assumptions:

1. **Stable Internet Connection:** It is assumed that users will have a reliable internet connection for seamless platform usage.

2. **Browser Compatibility:** Users are expected to access the platform through modern browsers (Chrome, Firefox, Safari etc).

3. **Availability of Third-Party APIs:** The availability and stability of third-party APIs for feature code execution.

Dependencies:

- **External Code Execution Service:** The project depends on a reliable external service for executing user code.
- **Challenge Evaluation Module:** Dependencies on a robust system for evaluating and scoring user-submitted challenge solutions.
- **Database Service:** Rely on a stable database service for storing and retrieving user projects.

These assumptions and dependencies play a crucial role in shaping the design and functionality of the coding platform. Any changes or inaccuracies in these assumptions could impact the project's development and operation.

Chapter 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface contains six major pages. All of these pages have different layouts according to the function they provide.

- **SignIn Page:** The user enters their credentials to enter their account.
- **SignUp Page:** The user provides information to join the service.
- **Home Page:** Here, users can see all their projects. They can create new projects or edit existing ones. Users can also see the languages they have tried coding challenges with.
- **Questions Page:** A list of questions is displayed that a user can attempt.
- **Challenge Page:** Here, the user can solve the challenge.
- **Editor Page:** The editor is displayed here.

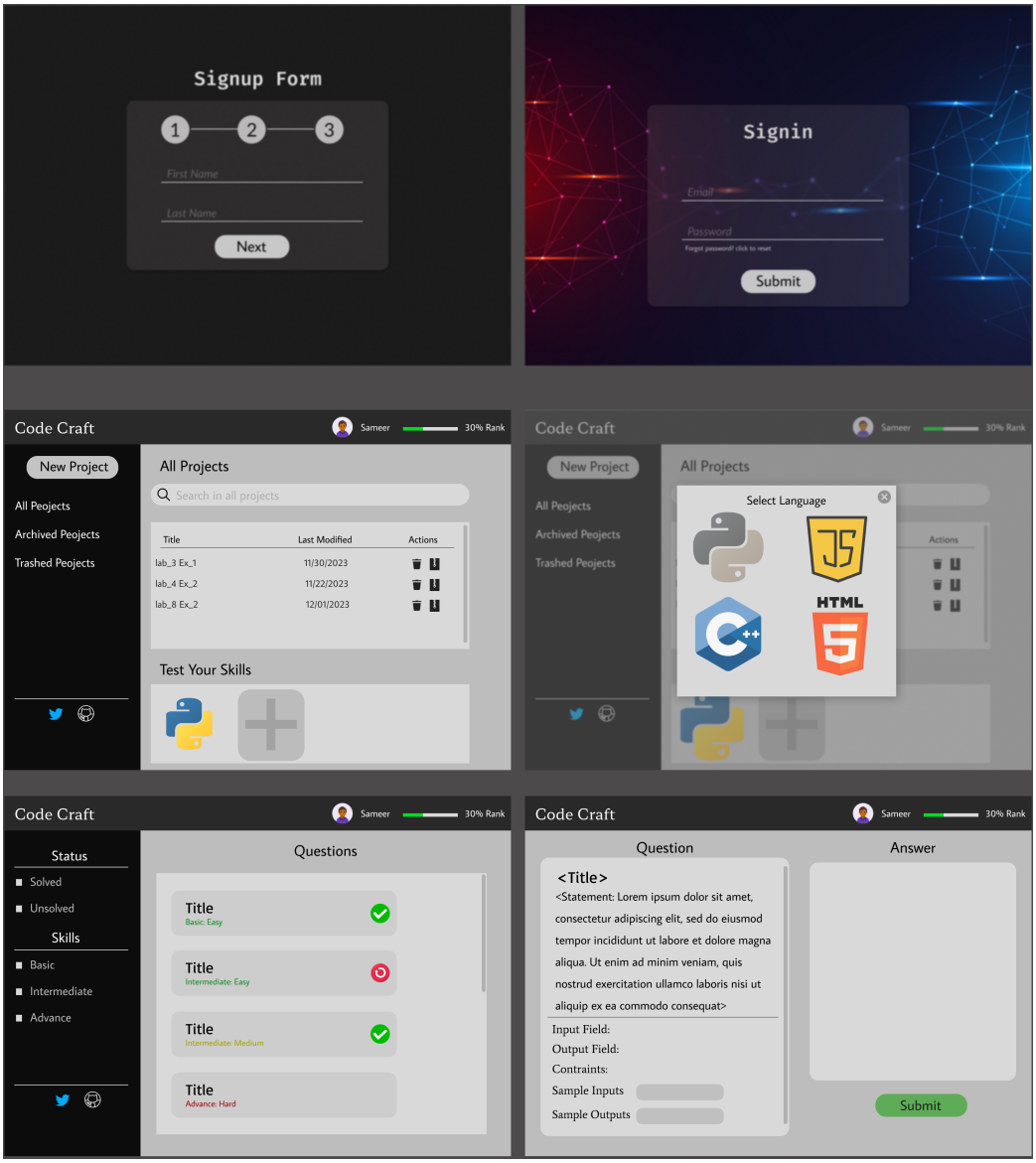

Figure 3.1: User Interface

3.1.2 Hardware Interfaces

Not applicable to this project

3.1.3 Software Interfaces

- For the database server, we have used SQLite 3.
- Our website will run on every operating system and mobile device if they have a web browser.
- For UI design, we have used Figma, and its implementation is done on React.
- For the code editor service, we have used CodeMirror, which is a JavaScript component that provides a code editor in the browser. It has a rich programming API and a focus on extensibility.

3.1.4 Communications Interfaces

CodeCraft is an online web service, so the user needs to have a web browser, and thus internet access is required. So it follows the HTTP communication standard. Passwords in the database will be encrypted using the PBKDF2 standard.

3.2 Functional Requirements

3.2.1 User Information

- User should be able to sign in to their account.
- User should be able to register via the sign-up form.

3.2.2 Code Editor Interface

- User should be able to access a user-friendly editor interface.
- The editor should provide support for multiple programming languages for execution.
- The editor should provide syntax highlighting according to the selected programming language.

- The editor should provide line numbering, auto-completion, and error highlighting.

3.2.3 User Projects

- User should be able to create,edit, save, and delete their projects.
- User should be able to execute their code and receive the output.
- User should be able to download a copy of their project on their local machine.

3.2.4 Coding Challenges Interface

- The user should be able to view and accept the challenges.
- Challenges should be of varying difficulty at different levels of expertise.
- The user should be able to select the programming language of their choice to complete the challenges.
- The user should receive points for the correct solution based on the difficulty level.

3.3 Behaviour Requirements

3.3.1 Use Case View

- Actors
  - The user using the application
- Main Page Use Case
  - The user accesses the application through this page.
  - Here, they are provided with two paths. Starting a coding project or starting with a coding challenge
- Registration Use Case
  - New user make account on this page to register for the service
- Project Use Cases

- A user can create new projects.
- A user can edit, delete, and archive an existing project

- Coding Challenges Use Cases
  - A user can select the programming language
  - A user can view, select, and solve a challenge

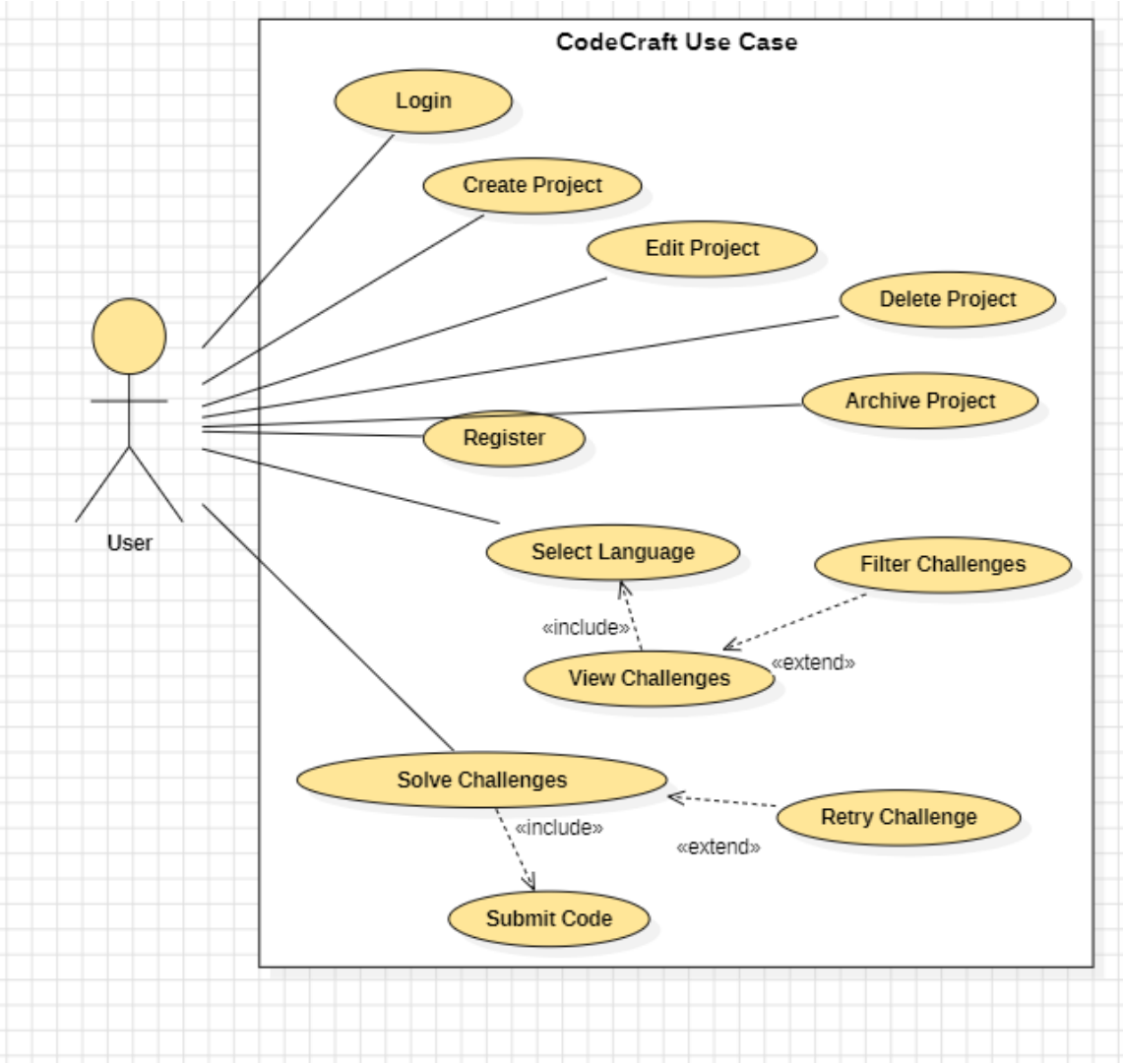

Figure 3.2: Use Case Diagram

Chapter 4

Other Non-Functional Requirements

4.1 Performance Requirements

CodeCraft is an interactive web application, so its performance is measured by its faster response time.

4.1.1 Responsive Design

The user interface should not take more than 10 seconds to load.

4.1.2 Code execution

The code should not take an excessive amount of time to execute.

4.1.3 Verification Time

After the user enters their credentials, whether as a registered customer or as a new customer, the time taken to verify the information should not exceed 10 seconds.

4.1.4 Download time

The time taken to download the code file should not be excessive.

4.2 Safety and Security Requirements

4.2.1 Safety

- end user should not use the website while driving, operating machinery, or in any other situation in which their focus is imperative.
- using the website for too long may cause strain to the eyes.

4.2.2 Security

- system should be secure enough so that all personal information, including email IDs and passwords, may not be disclosed unauthorized.
- developer will maintain the confidentiality of all user data.
- password should be encrypted to avoid unauthorized access.
- website should be malware-free to avoid damage to the end-user local system.

4.3 Software Quality Attributes

4.3.1 Scalablity

The system must possess features capable of adapting to a burgeoning user base and effectively managing a rising volume of code files and test attempts. This ensures that users can work efficiently, collaborate seamlessly, and sustain productivity as their projects undergo evolution and increase in complexity.

4.3.2 Compatibility

The website should be compatible with popular web browsers such as Chrome, Firefox, etc. It should be responsive and accessible on different devices, including desktops, tablets, and smartphones.

4.3.3 Reliability

The website should be available and reliable, with minimal downtime for maintenance.

4.3.4 Usability

The user interface should be intuitive and user-friendly. Clear instructions and guidance should be provided for using the code editor and taking skill tests.

4.3.5 Maintainability

The code base should be well organized and maintainable for future updates and enhancements. Regular updates and bug fixes should be carried out efficiently.

4.3.6 Robustness

Constraints on inputs are applied to ensure valid data is entered into the system.