

Darshan University

A Project Report on

**“Code Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – V

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Ehsaas Chaudhary | Enrollment No: 21010101028 |
| Academic Year  (2023-2024) | |
| Internal Guide  Prof. Ekta Baldha  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2101CS503)** for entitled **“Code Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of **Prof. Ekta Baldha** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Ehsaas Chaudhary

Student’s Signature

Date: 05/10/2023

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Code Management System”** has been satisfactorily prepared by **Ehsaas Chaudhary** (**21010101028**) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2023-2024.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. Ekta Baldha  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide **Prof**. **Ekta Baldha** and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Ehsaas Chaudhary**

**ABSTRACT**

In the ever-evolving landscape of computer science education, our comprehensive online code management platform emerges as a transformative solution for students, educators, and administrators. Rooted in a user-friendly interface, this platform seamlessly integrates a suite of powerful features designed to enhance the teaching and learning of coding within the academic realm.

For students, it provides an unprecedented avenue for collaborative coding, enabling real-time co-editing and version control. Automated code analysis and peer code reviews empower students to improve their coding skills while fostering a culture of constructive feedback and shared learning. A repository of code templates and snippets expedites the initiation of coding assignments, ensuring adherence to industry best practices. Robust code discussion forums and chat functionality offer students the opportunity to seek assistance and engage in code-related discussions effortlessly.

Educators find in this platform an indispensable ally, with tools such as assignment templates, efficient grading mechanisms, and plagiarism detection algorithms that uphold academic integrity. Detailed analytics and reporting features enable instructors to gain valuable insights into student performance and progress, facilitating data-driven decision-making in the classroom. Integrated teaching materials, including lecture notes and resources, further enrich the learning experience.

Administrators benefit from streamlined user management, role-based access control, and comprehensive usage analytics that ensure smooth platform operation. Automated notifications keep all stakeholders informed, and robust backup and recovery mechanisms prevent data loss, safeguarding valuable academic work.

Furthermore, the platform transcends conventional boundaries with a mobile app for on-the-go access, gamification elements for enhanced motivation, and support for multiple programming languages. Accessibility features align with WCAG standards, promoting inclusivity. Seamless integration with Learning Management Systems (LMS) widely adopted in educational institutions guarantees compatibility and facilitates the transition into modern coding education.

Top of Form

**Table of Contents**

[List of Figures I](#_Toc147485298)

[List of Tables II](#_Toc147485299)

[1 Introduction 1](#_Toc147485300)

[1.1 Product perspective 1](#_Toc147485301)

[1.2 Product features 1](#_Toc147485302)

[1.2.1 There are two different users who will be using this product: 1](#_Toc147485303)

[1.3 Functional Requirement 1](#_Toc147485304)

[1.3.1 Student 1](#_Toc147485305)

[1.3.2 Guides 2](#_Toc147485306)

[1.4 Non-Functional Requirement 3](#_Toc147485307)

[2 Design and Implementation Constraints 4](#_Toc147485308)

[2.1 Use case diagram 4](#_Toc147485309)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc147485310)

[2.3 Sequence diagram 7](#_Toc147485311)

[2.4 State diagram 8](#_Toc147485312)

[2.5 Class diagram 9](#_Toc147485313)

[2.6 Data flow diagram 10](#_Toc147485314)

[2.6.1 Context diagram (level-0) 10](#_Toc147485315)

[2.6.2 DFD Level-1 11](#_Toc147485316)

[3 External interface requirement (Screens) 12](#_Toc147485317)

[3.1 Screen-1: Profile Form 12](#_Toc147485318)

[3.2 Screen-2: project upload 13](#_Toc147485319)

[3.3 Screen-3: feedback 14](#_Toc147485320)

[4 Database design 15](#_Toc147485321)

[4.1 List of Tables 15](#_Toc147485322)

[5 Stories and Scenario 16](#_Toc147485323)

[5.1 Story-1: Register a User 16](#_Toc147485324)

[5.1.1 Scenario# S1.1 16](#_Toc147485325)

[5.1.2 Scenario# S1.2 16](#_Toc147485326)

[5.1.3 Scenario# S1.3 17](#_Toc147485327)

[5.2 Story-2: Search 17](#_Toc147485328)

[5.2.1 Scenario# S2.1 17](#_Toc147485329)

[5.2.2 Scenario# S2.2 17](#_Toc147485330)

[5.3 Story-3: Feedback 18](#_Toc147485331)

[5.4 Story-4: Upload code 18](#_Toc147485332)

[5.5 Story-5: Manage profile 18](#_Toc147485333)

[5.6 Story-5: Evaluate Code 18](#_Toc147485334)

[5.7 Story-5: Grade the output 18](#_Toc147485335)

[6 Test cases 19](#_Toc147485336)

[7 References 23](#_Toc147485337)

# List of Figures

[Figure 2.1‑1 Use case diagram for code management system 4](#_Toc147476749)

[Figure 2.2‑1 Activity diagram for code upload 5](#_Toc147476750)

[Figure 2.2‑2 Swimlane diagram for code upload 6](#_Toc147476751)

[Figure 2.3‑1 Sequence diagram for code upload 7](#_Toc147476752)

[Figure 2.4‑1 State diagram of code 8](#_Toc147476753)

[Figure 2.5‑1 Class diagram for code management system 8](#_Toc147476754)

[Figure 2.6‑1 Context diagram for code management system 9](#_Toc147476755)

[Figure 2.6‑2 DFD level-1 for code management system 10](#_Toc147476756)

[Figure 3.1‑1 Screen-1: Profile 11](#_Toc147476757)

[Figure 3.2‑1 Screen-2: Project Upload 12](#_Toc147476758)

[Figure 3.3‑1 Screen-3: feedback 13](#_Toc147476759)

# List of Tables

[Table 3.1‑1 Screen element of Profile 11](#_Toc147476743)

[Table 3.2‑1 Screen element of project upload 12](#_Toc147476744)

[Table 3.3‑1 Screen element of feedback 13](#_Toc147476745)

[Table 4.1‑4 Table: Admin 14](#_Toc147476746)

[Table 4.1‑3 Table: Code 14](#_Toc147476747)

[Table 4.1‑4 Table: Student 14](#_Toc147476748)

# Introduction

## Product perspective

Welcome to the "Code Management System." This innovative platform revolutionizes computer science education by offering students real-time collaboration, automated code analysis, and peer reviews. Educators gain tools for efficient assignment management, grading, and plagiarism detection. Administrators benefit from streamlined user management and comprehensive analytics.

This system goes beyond conventional boundaries with a mobile app, gamification elements, and multi-language support. It's committed to accessibility and easily integrates with Learning Management Systems (LMS). The "Code Management System" represents a transformative force, shaping the future of coding education in colleges and universities worldwide.

## Product features

### There are two different users who will be using this product:

* The features that are required for the Guide are:

1. Assignment Templates: Create assignment templates with criteria.
2. Grading Tools: Efficient grading and feedback mechanisms.
3. Plagiarism Detection: Ensure code originality.
4. Analytics and Reports: Track student performance and progress.
5. Integrated Teaching Materials: Upload teaching materials and resources.
6. User Management: Streamline user tasks (registration, activation, reset).
7. Role-Based Access Control: Customize user roles and permissions.
8. Usage Analytics: Track system usage and trends.
9. Automated Notifications: Implement automated alerts.
10. Backup and Recovery: Ensure data backup and recovery.

* The features that are required for the student are:

1. Code Collaboration: Real-time co-editing and version control for code assignments.
2. Code Analysis: Automated analysis with improvement suggestions.
3. Code Reviews: Peer code review capabilities.
4. Code Templates: Predefined templates for common programming tasks.
5. Code Discussion Forums: Chat and discussion forums.
6. Code Upload: Students can upload their code.
7. Profile Management: Users can manage their profiles.
8. Notifications: Receive notifications for updates.
9. Tagging: Add tags to code posts.
10. Voting and Comments: Vote and comment on others' code.
11. Search: Advanced search capabilities.

## Functional Requirement

### Student

* User Registration and Authentication:

1. Students must be able to create accounts with unique usernames and passwords.
2. They should be able to log in securely using their credentials.

* Profile Management:

1. Students should have the ability to update their profiles, including personal information and profile pictures.
2. They must be able to view and edit their contact information.

* Code Collaboration:

1. Students should be able to create new code assignments or join existing collaborative coding sessions.
2. Real-time co-editing of code with version control is essential.
3. The system must allow students to invite others to collaborate on their code.

* Code Submission and Analysis:

1. Students must be able to submit their code assignments.
2. The system should perform automated code analysis, providing suggestions for improvement.
3. Feedback and analysis results should be presented to the students.

* Code Reviews:

1. Students should be able to participate in peer code reviews.
2. They must have the ability to review and provide feedback on code submissions from their peers.
3. The system should facilitate discussions and comments during code reviews.

* Code Upload:

1. Students must be able to upload their code files for assignments and projects.
2. File formats, size limits, and version control should be managed by the system.

* Notifications:

1. Students should receive notifications for important system events, such as code reviews, feedback, or assignment deadlines.
2. Notifications can be sent via email or within the platform.

* Tags, Voting, and Comments:

1. Students should be able to add tags to their code posts for categorization.
2. They should be able to vote on others' code submissions.
3. Commenting on code posts should be allowed, promoting discussion and collaboration.

* Search Capability:

1. The system should support advanced search capabilities, allowing students to find specific code examples, discussions, and topics efficiently.

### Guides

* User Management:

1. Instructors should have the capability to manage student and teaching assistant user accounts within their courses.
2. This includes user registration, activation, deactivation, and password reset.
3. Instructors can assign roles and permissions to users based on their responsibilities within the course.

* Teaching Materials Management:

1. Instructors should have the ability to organize and categorize teaching materials efficiently.
2. They can update, delete, or add new teaching resources and lecture notes as needed.

* Assignment Analytics:

1. Instructors need access to detailed analytics for each assignment.
2. Analytics should include submission statistics, performance distributions, and trends over time.

* Assignment Versioning:

1. Instructors can maintain multiple versions of assignments and assignment templates.
2. This allows for iterative improvements and flexibility in course content.

* Assignment Templates Customization:

1. Instructors can customize assignment templates to fit the specific requirements of their courses.
2. Customization may include grading rubrics, assignment instructions, and resources.

* Assignment Feedback Tracking:

1. Guides can track the feedback provided to students on their assignments.
2. This helps in monitoring the quality and consistency of feedback across assignments.

* Assignment Scheduling:

1. Instructors can schedule assignment releases, deadlines, and reminders for students.
2. This ensures effective course planning.

* Grading Automation:

1. The system should support automated grading for assignments with predefined criteria.
2. Guides can review and modify automated grades as needed.

## Non-Functional Requirement

* Usability:

1. The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

* Accuracy:

1. The data stored about the books and the fines calculated should be correct, consistent, and reliable.

* Availability:

1. The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system responds to the requests within two seconds or less.

* Maintainability:

1. The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

# Design and Implementation Constraints

## Use case diagram

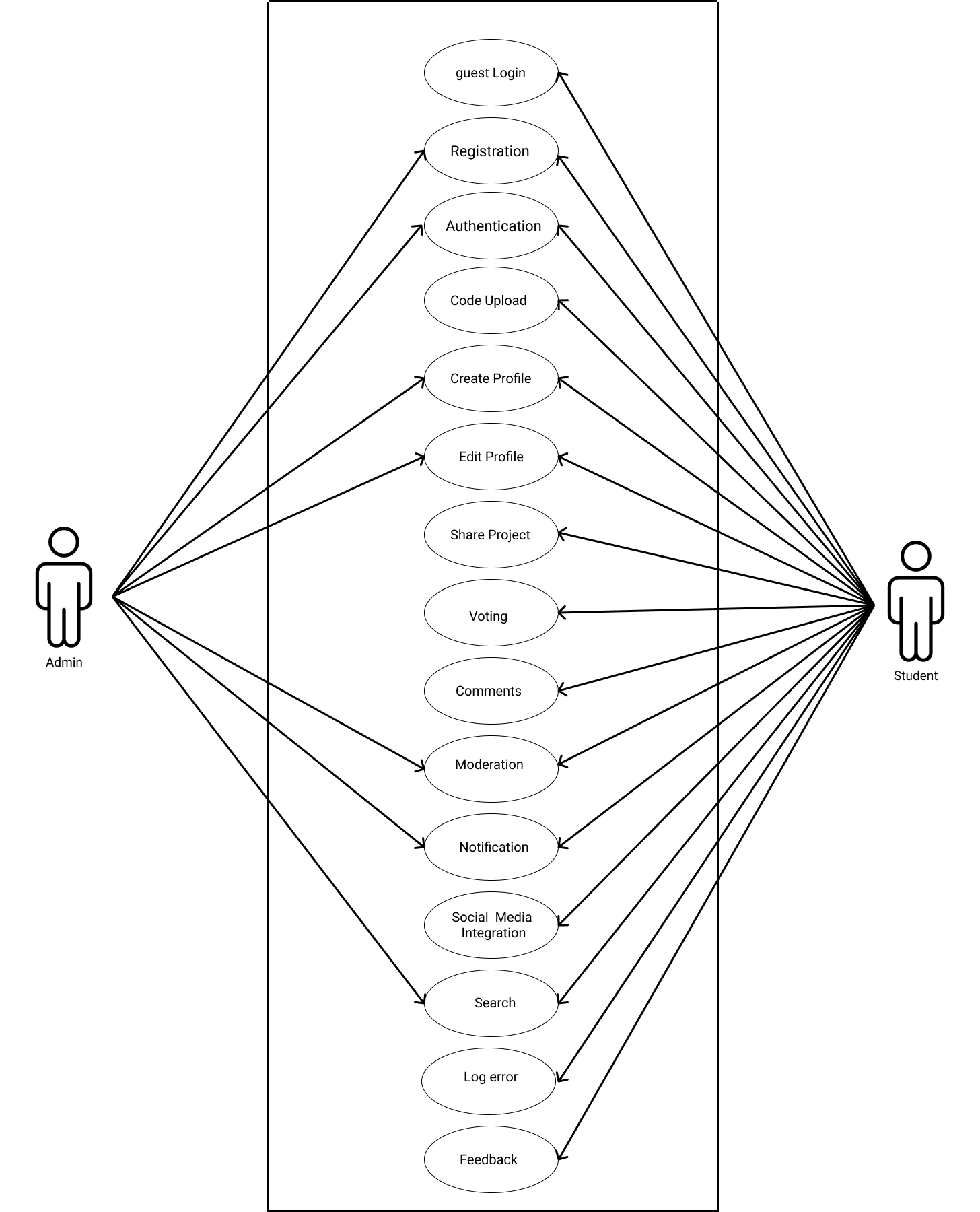


Figure 2.1‑1 Use case diagram for code management system

## Activity diagram and Swimlane diagram

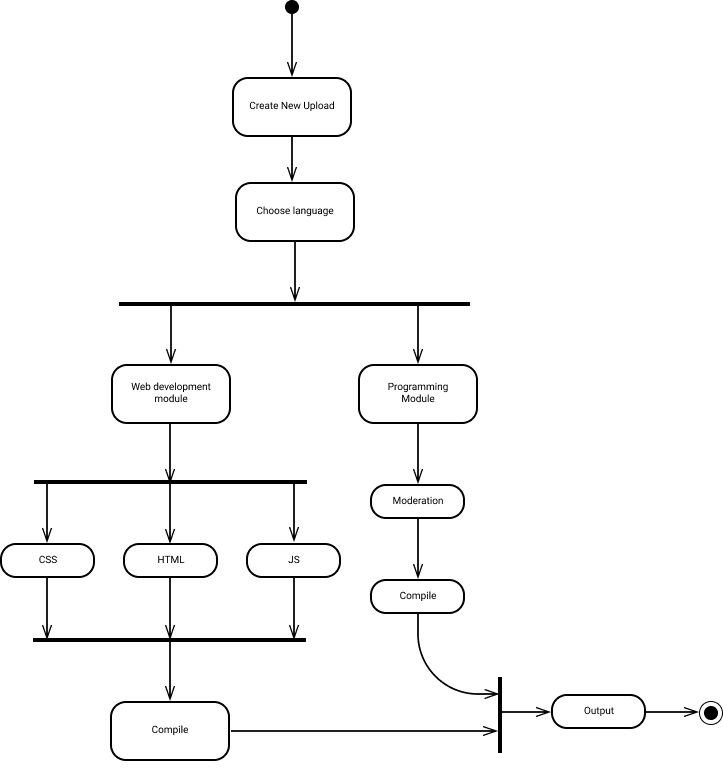


Figure 2.2‑1 Activity diagram for code upload

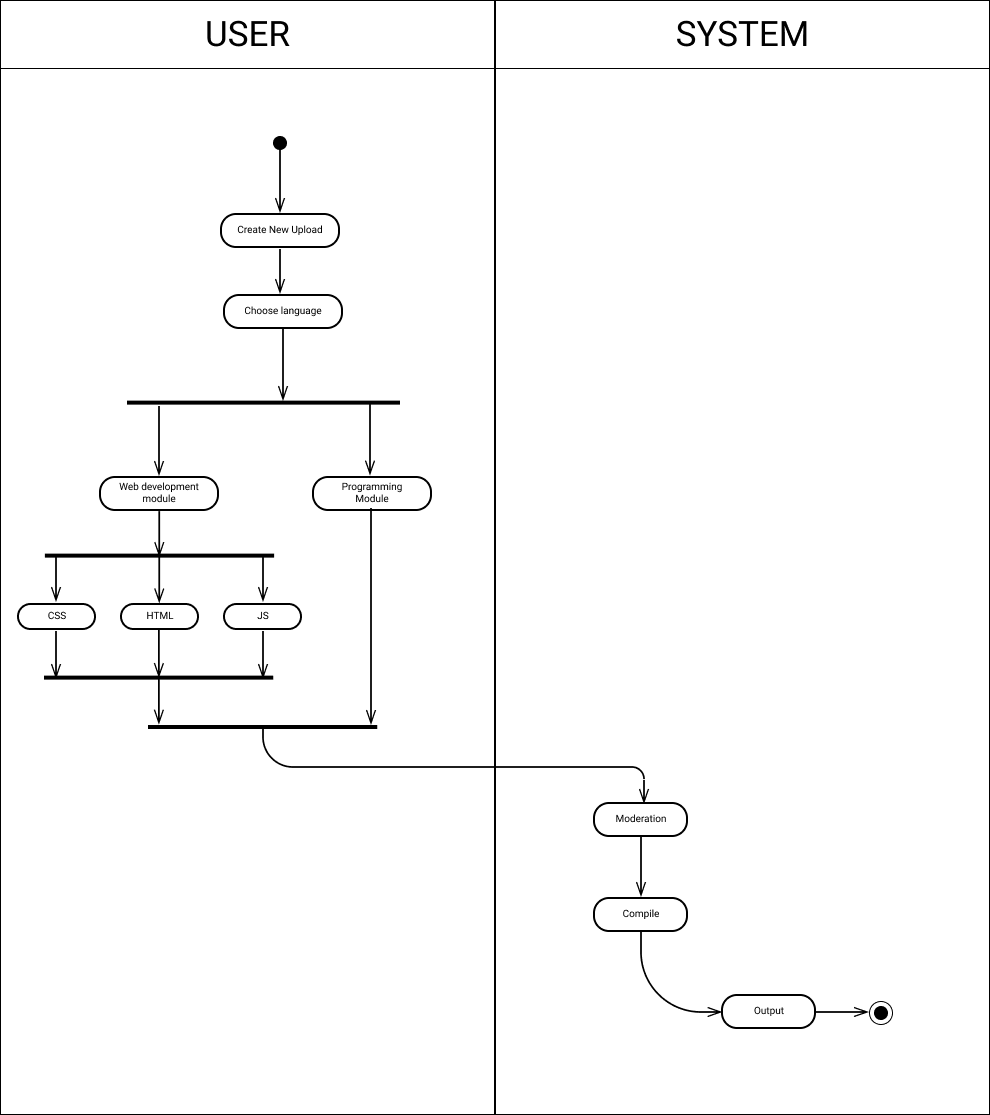


Figure 2.2‑2 Swimlane diagram for code upload

## Sequence diagram

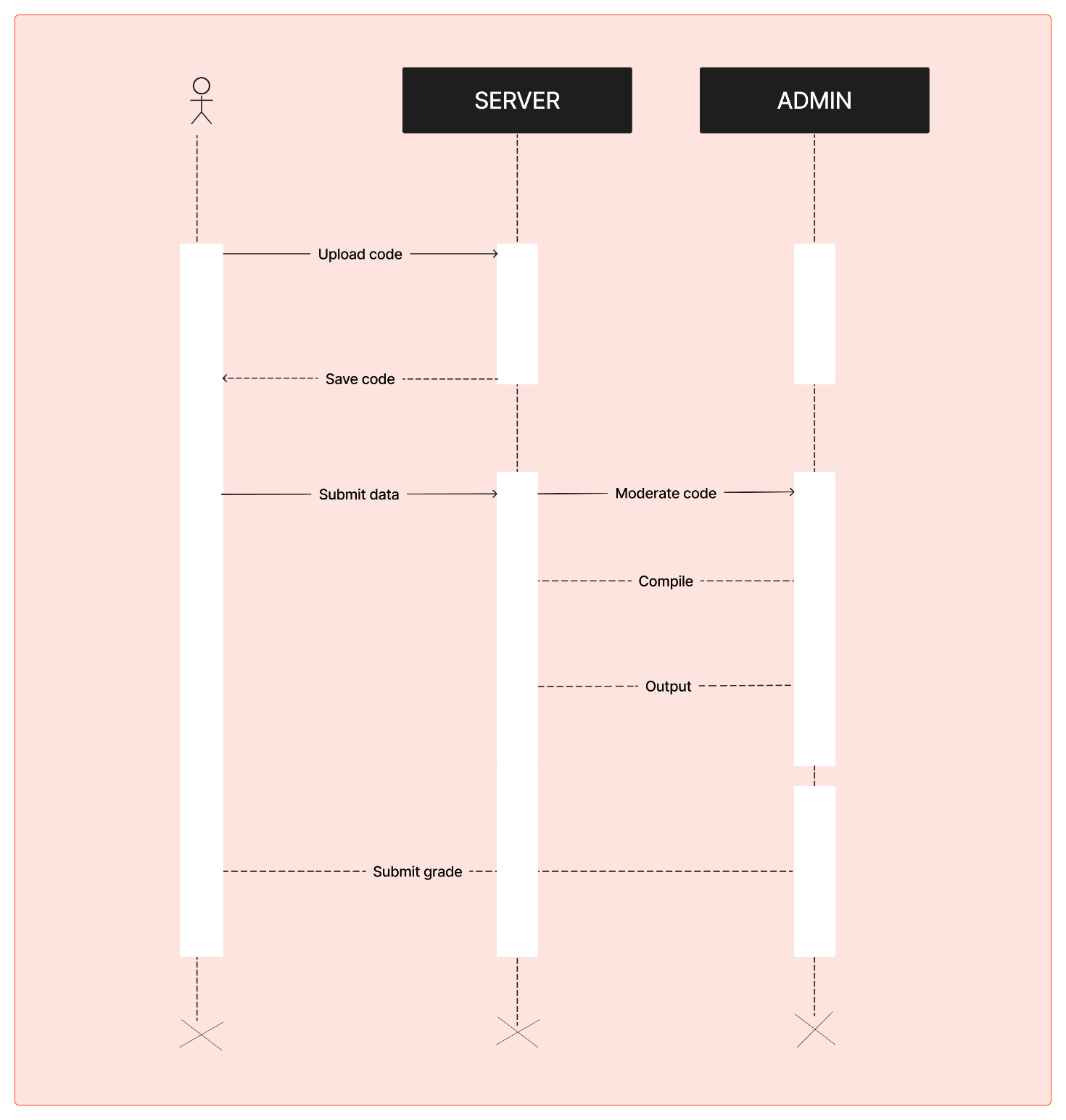


Figure 2.3‑1 Sequence diagram for code upload

## State diagram

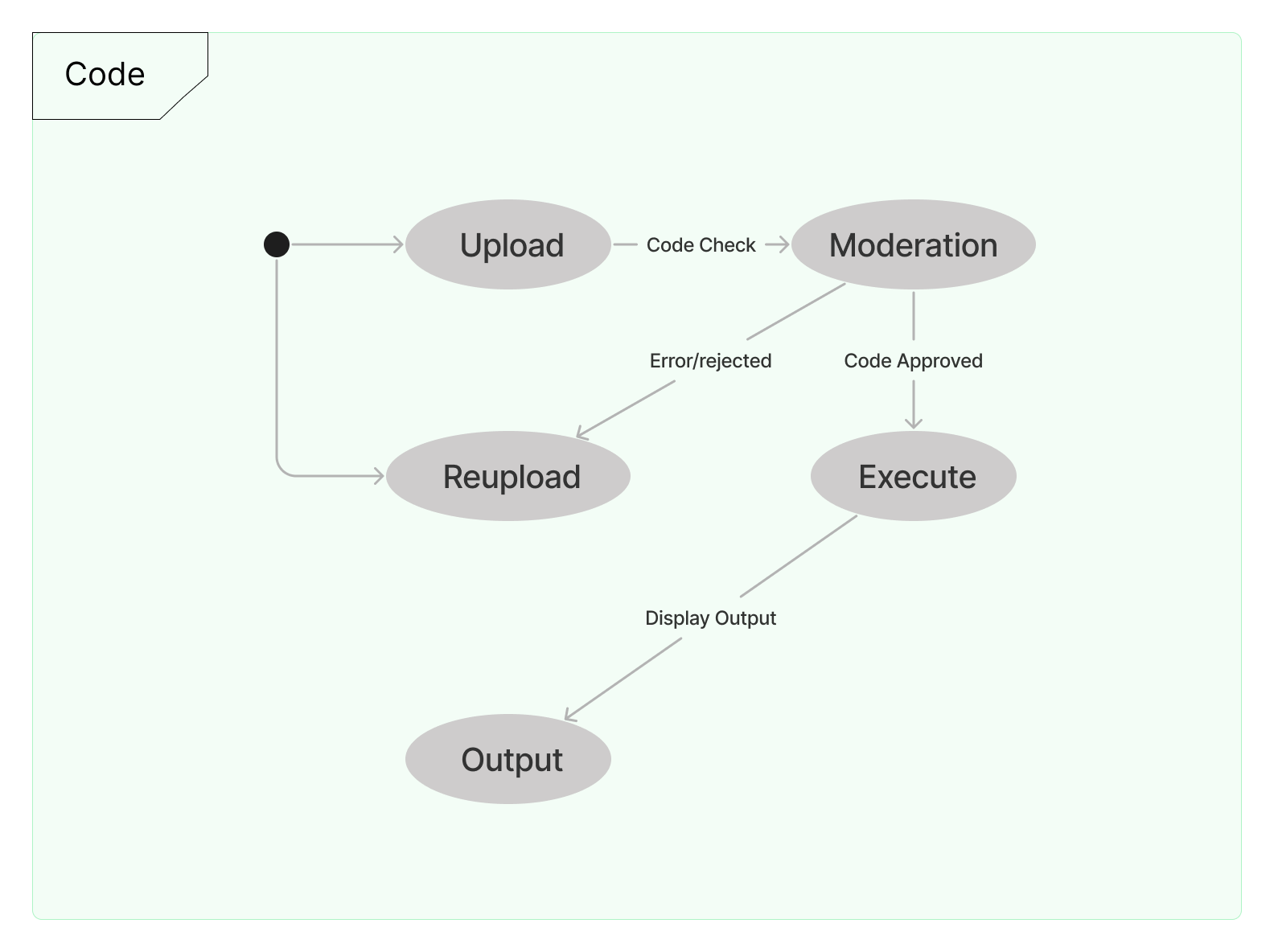


Figure 2.4‑1 State diagram of code

## Class diagram

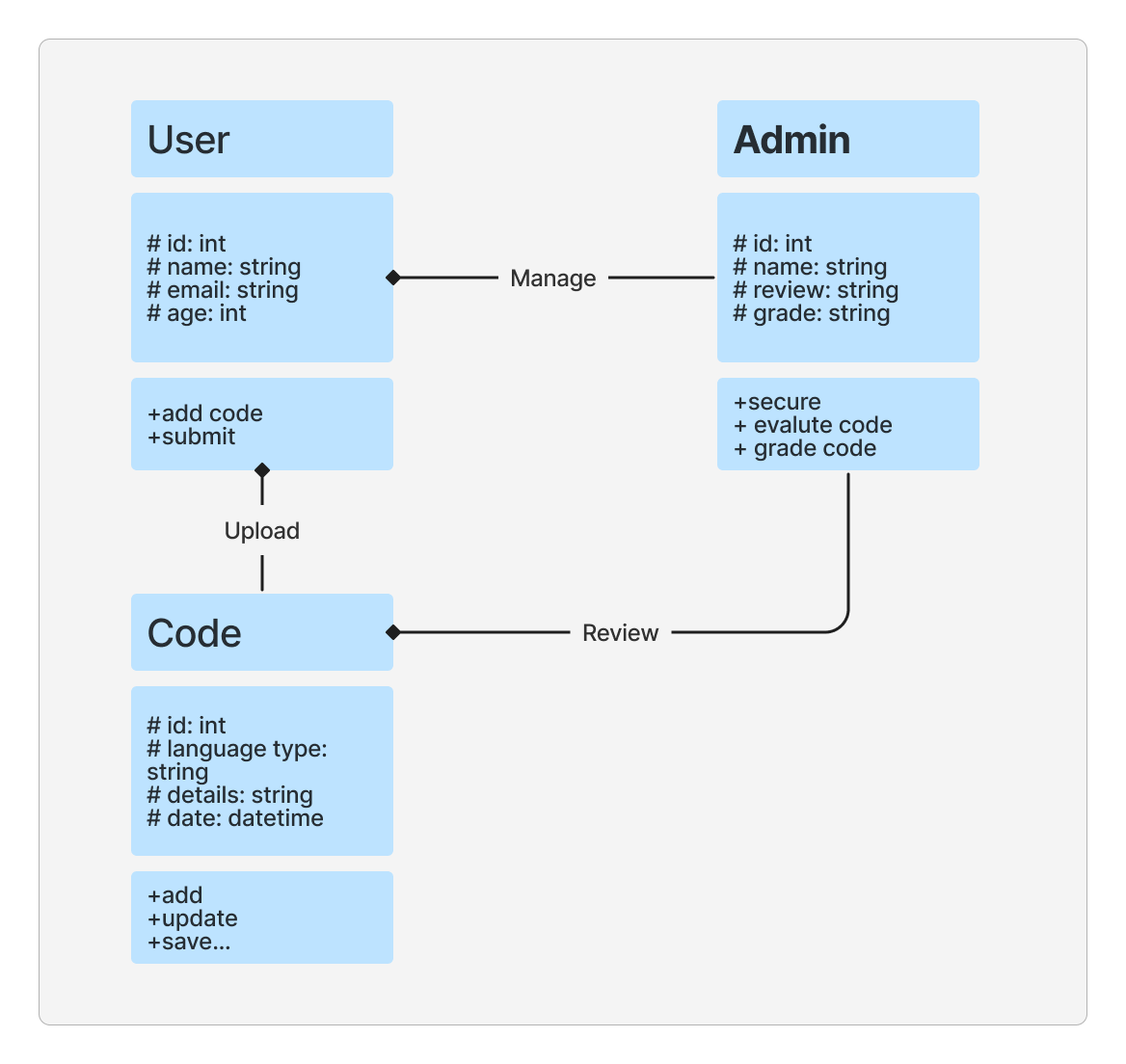


Figure 2.5‑1 Class diagram for code management system

## Data flow diagram

### Context diagram (level-0)

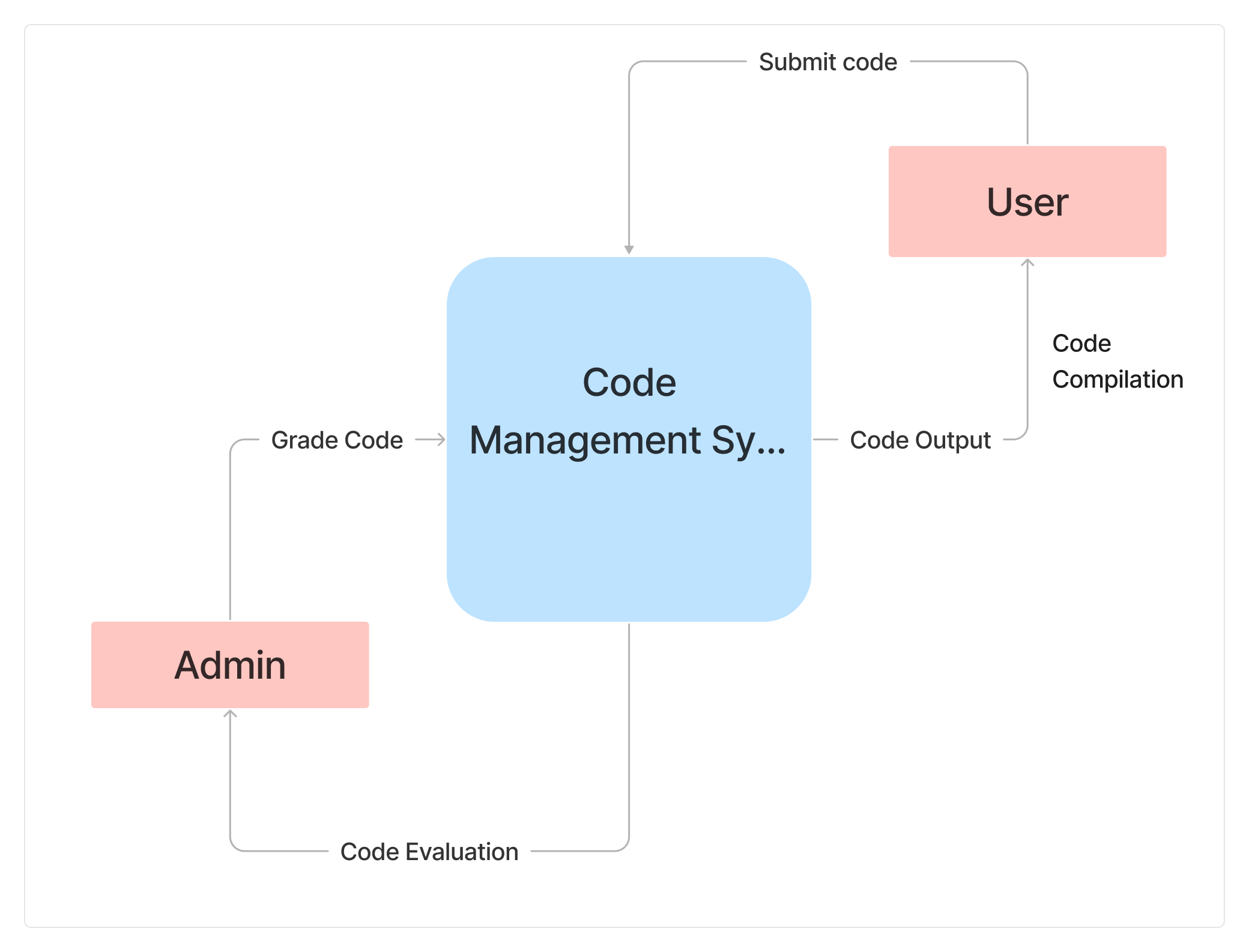


Figure 2.6‑1 Context diagram for code management system

### DFD Level-1

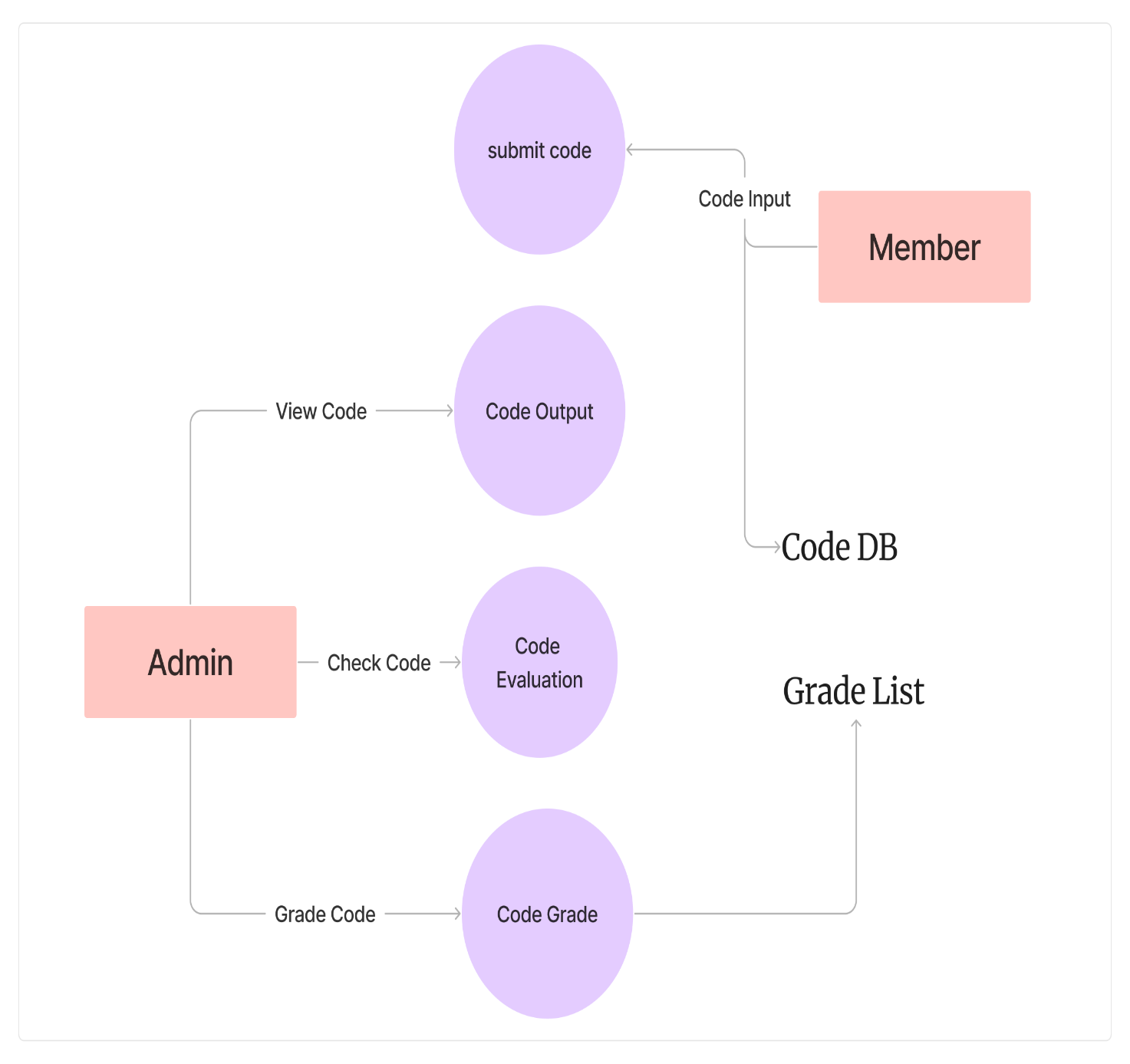


Figure 2.6‑2 DFD level-1 for code management system

# External interface requirement (Screens)

## Screen-1: Profile Form

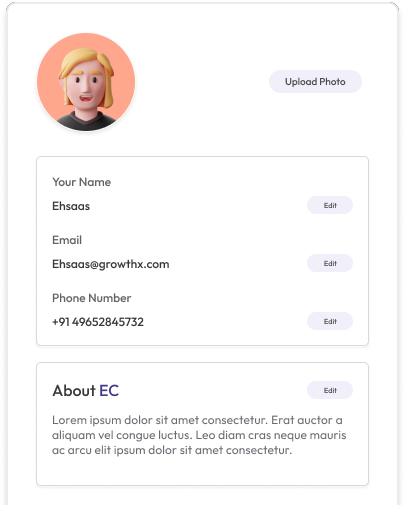


Figure 3.1‑1 Screen-1: Profile

**Purpose:** This page allow user to add a profile to there website page

Table 3.1‑1 Screen element of Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | contact | int | M | 1 | Contact field should be editable and accept only 10-digit mobile number. |
| 4 | About | Textbox | M | 1 | About field should be compulsory and accept at least 150 words |

## Screen-2: project upload

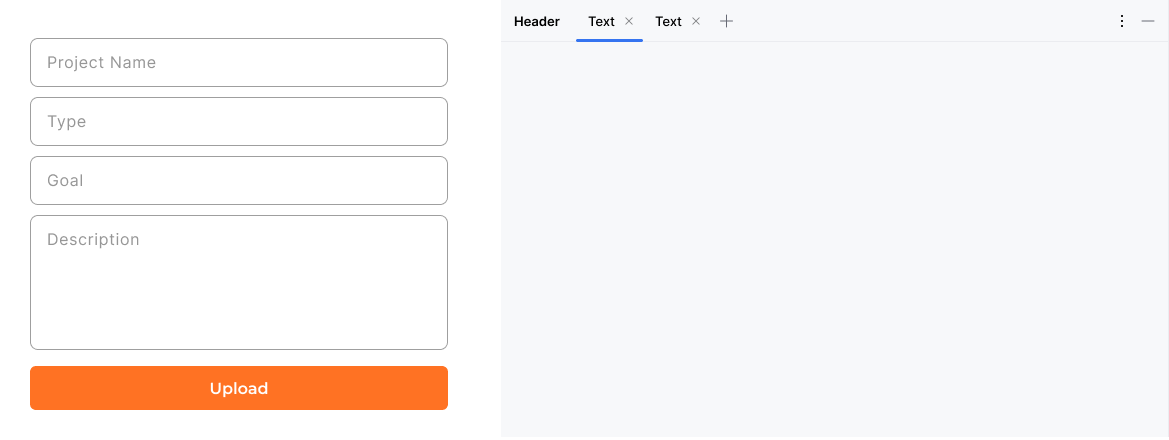


Figure 3.2‑1 Screen-2: Project Upload

**Purpose:** This module will allow the user to upload there project with a detailed description and necessary details

Table 3.2‑1 Screen element of project upload

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Project name | Textbox | M | 1 | Project Name field should be editable and accept the Username. |
| 2 | Type | Textbox | M | 1 | Type field should be editable and accept the type in bold letters |
| 3 | Goal | Textbox | M | 1 | Goal field should be compulsory and accept at least 150 words |
| 4 | Description | Textbox | ------ | ------ | Description field should be compulsory and accept at least 300 words |

## Screen-3: feedback

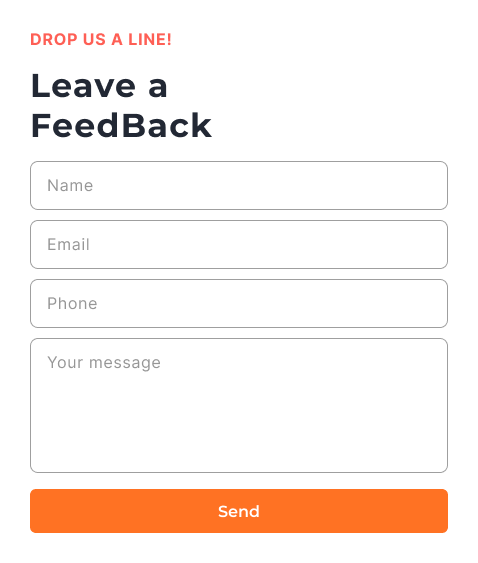


Figure 3.3‑1 Screen-3: feedback

**Purpose:** This module will allow the user leave feedback for the developer and admin

Table 3.3‑1 Screen element of feedback

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full Name | Textbox | M | 1 | First Name field should be editable and accept only string as a borrower first name. |
| 2 | Email | Textbox | O | 1 | Email should be editable and accept only string with proper format |
| 3 | Message | Textbox | M | 1 | Message field should be editable and accept max 250 words |
| 4 | Contact | Textbox | M | 1 | Contact field should be editable and accept only 10-digit mobile number. |

# Database design

## List of Tables

* Admin
* Code
* Student

Table 4.1‑4 Table: Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AdminID | int | NN | PK (Auto Increment) |  |
| AdminName | varchar(100) | AN |  |  |
| Gender | varchar(100) | AN |  |  |
| DOB | DateTime | AN |  |  |
| Department | varchar(100) | AN |  |  |
| Contact | number(10,0) | AN |  |  |

Table 4.1‑3 Table: Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CodeID | int | NN | PK (Auto Increment) |  |
| Language Type | varchar(100) | AN |  | C programming |
| Output Id | int | AN |  |  |
| Created | DateTime | AN |  |  |

Table 4.1‑4 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StudentID | int | NN | PK (Auto Increment) |  |
| StudentName | varchar(100) | AN |  | Ehsaas Chaudhary |
| Gender | varchar(100) | AN |  |  |
| DOB | DateTime | AN |  |  |
| Department | varchar(100) | AN |  |  |
| Contact | number(10,0) | AN |  |  |

# Stories and Scenario

## Story-1: Register a User

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a user,  I want to sign up and register myself  So that a I can use the website efficiently and save all my data. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The signup page and registration functionality are most imp for working of a website as the main purpose of this website is severing user an easy way to manage their code and share it with easy with easy user profile management. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | User enters all the correct info on the sign-up page and is a new user accessing the website for first time |
| Prerequisite | **:** | Webpage of sign up should be launched properly |
| Acceptance Criteria | **:** | **Given:** user navigate to the sign-up page.  **When:** User uses correct name format, correct email address and correct password.  **Then:** user will be navigated to the home screen  Display username detail in profile  Display home page per content.  Display the data in correct format.  Save all the data in database.  Proceed with using website and its feature |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | User enters all the incorrect info on the sign-up page and is a new user accessing the website for first time |
| Prerequisite | **:** | Webpage of sign up should be launched properly |
| Acceptance Criteria | **:** | **Given:** user navigate to the sign-up page.  **When:** User uses incorrect name format, incorrect email address and incorrect format of password.  **Then:** user will be stay to the sign-up screen  show popup message for the wrong name entry.  show popup message for the wrong email.  show popup message for wrong format for password.  Ask to fill in captcha details  Disable the sign-up button at the end of the form. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | User enters all the empty info on the sign-up page and is a new user accessing the website for first time |
| Prerequisite | **:** | Webpage of sign up should be launched properly |
| Acceptance Criteria | **:** | **Given**: user navigate to the sign-up page..  **When**: User Click “sign-up” button.  **Then** :  show error message for “Enter username”.  And show error message for “Enter email address”.  And show error message for “Enter password”.  And show error message for “Enter captcha digits”.  And show error message for “enter all details” when button is hovered. |

## Story-2: Search

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a user,  I want to search  So that I can view publicly available projects. |
| Priority | **:** | High |
| Estimate | **:** | L |
| Reason | **:** | The search function is one of many ways to view projects and can come very handy as times |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.1* | : | User uses correct name for the search function |
| Prerequisite | **:** | Homepage should be launched properly |
| Acceptance Criteria | **:** | **Given:** User and Guest navigate to the Home page.  **When:** The user is entering some project name in the search bar  **Then:** user will be navigated to the project screen  Display project detail  Display project owners name.  Display project info.  Display project related comments.  Display share option.  Display technologies used |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.2* | : | User uses incorrect name for the search function |
| Prerequisite | **:** | Homepage should be launched properly |
| Acceptance Criteria | **:** | **Given:** User and Guest navigate to the Home page.  **When:** The user is entering some project name in the search bar  **Then:** user will not be navigated to the project screen  Don’t display project detail  Don’t display project owner’s name.  Don’t display project info.  Don’t display project related comments.  Don’t display share option. |

## Story-3: Feedback

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As a user or guest,  I want to be able to provide feedback  So that I can communicate my thoughts to the admin/owner. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | The feedback function is a very useful ability to get users view and get your website to a better level. |

## Story-4: Upload code

|  |  |  |
| --- | --- | --- |
| *Story # S4* | : | As a user,  I want to be able to submit my code and view its output,  So that I can save mu code and share it further. |
| Priority | **:** | High |
| Estimate | **:** | XXL |
| Reason | **:** | The ability to submit code on the go is the key feature of the website one which should not be ignored and must work at all cost |

## Story-5: Manage profile

|  |  |  |
| --- | --- | --- |
| *Story # S6* | : | As an user,  I want manage my profile page,  So that I can save my info and express myself to all the visitor who visit my page. |
| Priority | **:** | Medium |
| Estimate | **:** | XL |
| Reason | **:** | Ability to have a profile in a website is very imp for the user experience and for the data that website has to handle. |

## Story-5: Evaluate Code

|  |  |  |
| --- | --- | --- |
| *Story # S7* | : | As an admin,  I want to be able to open and check the code of the users,  So that I can make sure that it doesn’t break and law or rules. |
| Priority | **:** | medium |
| Estimate | **:** | XL |
| Reason | **:** | The security feature for content that will be posted in the site is a imp one to prevent people from submitting bad code. |

## Story-5: Grade the output

|  |  |  |
| --- | --- | --- |
| *Story # S8* | : | As an admin,  I want to be able to grade the code from A to D,  So that I can score the project on the basis of the evaluation. |
| Priority | **:** | Low |
| Estimate | **:** | S |
| Reason | **:** | Grading is not compulsory as it is done my AI too admin is needed to make prompts work properly. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | CodeHub | Test Designed by: | Ehsaas Chaudhary |
| Module Name: | **FeedBack** | **Test Designed date:** | 25/08/2023 |
| Release Version: | **1.0** | **Test Executed by:** | **Ehsaas Chaudhary** |
| Test Case ID | **TC\_003** | **Test Execution date:** | 26/08/2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: FeedBack Webpage should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** |  |
| TC\_001 | Provide correct fields essential to fill the feedback form | Functional | Filling the feedback form with correct info |  |
| TC\_002 | Provide incorrect fields essential to fill the feedback form | Functional | Filling the feedback form with incorrect info |  |
| TC\_003 | Varify feedback form elements | GUI | Varify that all elements are availabe on feedback form |  |

|  |  |
| --- | --- |
| **Test Case Title** | Fill the feedback form with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | FeedBack Webpage should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Access feedback page URL | The page launched properly | page launched successfully | Pass |  | codeeditore/feedback |  |
| 2 | Enter valid User name in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Ehsaas Chaudhary |  |
| 3 | Enter valid Email in Email address field | Email field should be editable and accept the email address ending with gmail.com and have @ in it | email field should had @ some where inside the input and ended with gmail.com | pass |  | email: ec@gmail.com |  |
| 4 | Enter valid phone number in phone field | phone field should be editable and accept number/digits only | number added was correct | Pass | user should use appropriate country code with 10-digit | +91 1733121214 |  |
| 5 | Enter feedback message that you would like to pass on to the owner the site | message should be entered successfully wihtout any typo | message was passed on successfully | pass |  | All Test cases are up to date and correct |  |

|  |  |
| --- | --- |
| **Test Case Title** | Feedback to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Feedback Webpage should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to fill feedback with invalid Username | Should display an error message enter wrong username | Display an error of wrong username | Pass |  |  |  |
| 2 | Verify that User is not able to fill feedback with invalid email address | Should display an error message enter wrong email | Display an error of wrong email | Pass |  |  |  |
| 3 | Verify that User is not able to fill feedback with invalid phone number | Should display an error for number not found | Display an error phone number not found | Pass |  |  |  |
| 4 | Verify that User is not able to leave the feedback field blank | Should display an error for feild can not be empty | Display an error of empty field | Fail | Don’t Allow user to leave an empty field |  | Bug\_001 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify feedback page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Feedback webpage should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  |  |  |
| 2 | Verify that the feedback screen contains elements such as userName, Email, Phone number, feedback section and a send button | All listed control displayed properly on the page | feedback page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, mail, phone number has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |

# References

* <https://www.geeksforgeeks.org/software-requirement-specification-srs->
* <https://en.wikipedia.org/wiki/Software_requirements_specification>
* <https://medium.com/analytics-vidhya/what-is-srs-software-requirement-specification-document-and-its-requirements-b57bf87bb868>