Online Judge

SECURED ONLINE CODING PLATFORM (DJANGO-BASED WEBSITE)

Software Requirement Specification

Muskan (2020CSB1100)

Tummala Sanjana Reddy (2020CSB1137)

Kushal Agrawal (2020CSB1096)

Ruchika Sharma (2020CSB1119)

Isha Goyal (2020CSB1089)

Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms and Abbreviations
   4. Software stack
   5. References
   6. Overview
2. Overall Description
   1. Product Perspective
      1. System Interface
      2. User Interface
      3. Software Interfaces
      4. Communications Interfaces
      5. Site Adaptation Requirements
   2. Product Functions
   3. User Characteristics
   4. Constraints
   5. Assumptions and Dependencies
   6. Apportioning of Requirements
3. Specific Requirements
   1. External Interface
      1. Registration Interface
      2. Login Interface
      3. Super User Interface
      4. Setter Interface
      5. Participant Interface
   2. Functions
      1. Registration

3.2.1.1 Email Verification

* + 1. Login (Authentication and Authorisation)

3.2.2.1 Forgot Password

* + 1. Super User Mode
    2. Setter Mode
       1. Add Problem
       2. Delete Problem (if allowed)
       3. Modify Problem/Testcase (if allowed)
    3. Participant Mode
       1. Submit programs to problems (In 4 different languages)
       2. Verdict of every submission (AC/WA/TLE/RE)
       3. History of Submissions (Time stamp and Verdict)
       4. Leaderboard (Tracking progress)
    4. Personal Info
       1. Update Profile
       2. Change Password
  1. Performance Requirements
     1. Fairness
  2. Logical Database Requirements
  3. Software System Attributes
     1. Reliability
     2. Security
     3. Portability

1. About
2. Introduction

An Online Judge platform that compiles and runs codes to a problem securely, and judges if the submitted code passes the test cases or not within a given time limit. It will also provide access to the Leaderboard and the history of submissions to the user.

We will be using Python for implementation by exploring Docker techniques.It will be implemented using Python and Django libraries to handle some in-depth features of authentication, authorization and session maintenance.

It will remote run the code securely and gives the verdict.

* 1. Purpose

The purpose of this Software Requirements Specifications document is to specify the requirements of an online coding platform which is designed as a part of the course CS305 – Software Engineering. This document is intended to serve as a formal reference for platform coordinators to understand the functionalities of the software

* 1. Scope

This is an automated online judge platform. This project will cater to the needs of educational institutions and organizations, which want to give problems for practice to the users. If those problems were to be solved within the given time-frame for evaluation, then the time-stamp of submission can be used to evaluate them accordingly. But the system allows the user to anytime submit the solution to any problem persisting in the system.

The system will allow the participants to submit programs in 4 different languages (C++, C, Python, Java). The system provides access to participants to keep track of their progress by maintaining leaderboard. The participant is only allowed to see their own solutions.

It will allow the problem setters to add the problems, the corresponding testcases along with other constraints like time limit etc. It will also allow the later modification and deletion of problems for which you are authorized(i.e. Author of the problem can only modify the problem).

Superuser can do any modification in problems as well as is responsible for managing users and their roles in the system.

* 1. Definitions, Acronyms, and Abbreviations

DBMS: Database Management System

HTTPS: Hypertext Transfer Protocol Secure

HTML: Hyper Text Markup Language, a language in which the World Wide Web pages are written

CSS: Cascading Style Sheets, for adding style (e.g., fonts, colors, spacing) to Web documents.

JS: JavaScript, a lightweight object-oriented scripting language, that is used for developing HTML pages that are dynamic and have interactive effects on their web pages.

Docker: It provides the ability to package and run an application in a loosely isolated environment called a container and this allows many containers to run simultaneously on a given host.

SSR: Server-side rendering means that the content of your site is rendered on the server and not on your browser using javascript and then sent to the browser. It helps in putting less constraints on the client’s machine’s specifications and resources.

* 1. Software Stack

Python, Django: The general programming language used would be python, as it is versatile, easy to use and implementation is quicker.

HTML, CSS, JavaScript: These technologies will be used as we will be doing server-side rendering of web pages to make the product more accessible(weighing accessibility over scalability as problems are persistent in the system for the whole duration).

Docker: For isolation and security purpose, the codes will be compiled and executed in separate containers.

Git: Git is used to manage the source code, for tracking changes in the source code, enabling multiple developers to work together.

* 1. References

1. Software Engineering – A Practitioner’s Approach by Rogers Pressman, 5/e McGraw Hill Publications.
2. Clean Code – A Handbook of Agile Software Craftsmanship, Robert C. Martin
   1. Overview

The rest of the document is organized as follows:

Section 2 of the document lists the overall requirements of the software. This section gives a description of the user interface, software interface and the communication interfaces that will be a part of the product. The different modes of operations as well as the functionalities, constraints and dependencies of the software are sketched out.

Section 3 of the document talks about the specific requirements of the software, at length. Here, the external interfaces along with the different modes available are explained in detail. Other issues like performance and logical database requirements, design constraints and software attributes like security, and reliability are also discussed.

Section 4 gives the tools used to write the document.

1. Overall Description
   1. Product Perspective

This project is independent and self-contained. It is not a part of any larger software and can be used as a standalone product.

* + 1. System Interface

As this is not a part of any larger system, there are no system interfaces.

* + 1. User Interface

The product shall offer a web-based interface which can be operated in the following modes:

1. Super User mode
2. Setter mode
3. Participant mode

Further details in these modes can be obtained in section 3.2 of this document.

User interface:

* Navigation Bar to go to appropriate section.

Participant:

* Drop down menus for choosing programming Language.
* Text area to write solution program to be submitted.
* Submit button

Setter:

* Drop down Menu: To choose among various options in modifying problem.
* Event Listener : To extend form for that many testcases, Extending form based on the option chosen from drop down menu.

Super User:

* Default UI provided by Django for Admin.

Since the users of this software are assumed to have prior knowledge of online judging environment and its functionalities, the interface will be designed keeping in mind their technical background.

* + 1. Software Interfaces
* Docker: The system should have docker preinstalled on it to test the work.
* Web-Server: The server must support Java dynamic page views, standard network security protocols.
  + 1. Communications Interfaces

The web interface shall use the HTTPS protocol for secure communication with the server. The Window based client shall use the TCP/IP protocol for the same.

* + 1. Site Adaptation Requirements

The default settings shall be made based on standard configurations commonly used in most online judge platforms.

* 1. Product Functions

The major functionalities that the software shall provide are:-

* Authentication and Authorization.

1. Participant should not be allowed to access url concerned with admin and setters.
2. Setters should not be allowed to access admin’s urls and functionalities.

* Session Maintenance
* Enabling Problem setters to add/modify/ delete problems and testcases.
* Enabling participants to submit solutions.
* Evaluation of code in separate isolated environment using Docker(Safe and Secure)
* Enabling participants to view their past solutions
* Enabling participants to track their progress by maintaining leaderboard.
* Admin to maintain users and their roles and overall control on database.
* Participants/Setters should have option to update their profiles while their session is active.
* Participants/Setters should have option to change their password while their session is active.
* Email Verification while registration.
* Forgot Password and validating the user through email verification
  1. User Characteristics

The average user of this product is assumed to be familiar with online programming contests and their functioning. He/She is also expected to know standard terminology like ‘judge’ , ‘AC’ , ‘WA’, ‘TLE’ , ‘RE’ , ‘testcases’ , ‘submissions’ etc. We have optimized our interface to suit the needs of such a user.

* 1. Constraints

The web server should support JavaScript and Dynamic Forms rendering. We are using Server-Side Rendering for the front end as the problems are not part of any constrained time-framed contest and can be solved at any time, so making it available widely i.e. giving priority to accessibility over scalability.

Also, the server must ensure the safety and security of data and the authentication of user logins and passwords.

* 1. Assumptions and Dependencies

The server which runs the admin program needs to have the following softwares installed:

* Windows operating system
* Docker
  1. Apportioning of Requirements

The following requirements may not be part of the current release but will be included in the future releases:

* A plug-in for the judge program to detect code plagiarism and other foul practices.
* Facilitate customized searches for problems based on author , difficulty , date etc.
* Adding time-constrained contests.

1. Specific Requirements
   1. External Interface

The external interface shall provide a GUI system to interact with different categories of users in the system.

* + 1. Registration Interface

The system shall provide an interface by which participants can register themselves on the system. The user will be marked activate after email validation and confirmation only.

* + 1. Login Interface

The Login Interface shall allow a user (superuser/setter/participant) of the system to login to the system securely. It should provide the forgot password option using email check.

* + 1. Super User Interface

The default GUI provided by the Django to allow the administrator to control and organize users and problems.

* + 1. Setter Interface

It shall enable the setters of problems to add/modify/ delete problems and test cases for which they have permission.Update profile and change password option should be there in addition of this.

* + 1. Participant Interface

The interface shall allow the authenticated users to view problems, submit solutions, view verdict of the submissions, track the progress using leaderboard, history of submissions.Update profile and change password option should be there in addition of this.

* 1. Functions

The system shall include three basic modes of operation i.e. Super user mode, Setter mode, Participant mode. Detailed description of each phase is provided below:

* + 1. Registration

Every user who wishes to solve problems in the system shall first register to the system by providing the necessary details.

* The user shall fill in the required data.
* The user shall choose a unique username
* The user shall choose a strong password which he/she shall use during login.
* Multiple users with single mail is not allowed
  + - 1. Email Verification

The system forbids the individuals from registration with invalid email address as to get activated in the system, email verification is a must to do.

* + 1. Login (Authentication and Authorization)

Registered user can login by providing the correct credentials.

* + - 1. Forgot Password

If the registered user forgots the password then through email, the user should be able to reset the password.

* + 1. Super User Mode

The Django’s default administration’s functionalities are sufficient to cater the needs of this mode.

* + 1. Setter Mode

The problem setters will be provided the dashboard page where different options i.e. Add problem, Modify Problem, Delete Problem, Update Profile, Change Password will be given.

* + - 1. Add Problem

For adding Problem after adding the necessary data, then with the help of event listener feature of javascript, the form is getting extended according to the value entered as number of test cases, then in one go only the setter can add problem and the corresponding test cases for that.

* + - 1. Delete Problem (if allowed)

If the session’s user is the one who had added the problem to the system then he is allowed to delete that problem from the system also, otherwise no.

* + - 1. Modify Problem/Testcase (if allowed)

Three types of modifications are possible:

* Edit Problem Statement
* Edit Test Case
* Add Test Case

An event listener is associated with this field and more fields will get displayed to the user in accordance with the option chosen.Javascript will be used for the same.

* + 1. Participant Mode

The authenticated user will be allowed to view problems and submit solutions.

* + - 1. Submit programs to problems (In 4 different languages)

The participant will be given a dropdown menu from which he can choose the language in which he/she wants to submit the solution.

* + - 1. Verdict of every submission (AC/WA/TLE/RE)

The participant’s code should be compiled and run against the test cases in the secure isolated environment using docker. The various constraints like Time Limit can be passed as additional arguments to the container and accordingly it have to give 1 out of 4 verdicts i.e. AC, WA, RE, TLE. The verdict page will display the same.

* + - 1. History of Submissions ( Time stamp and Verdict)

The user will be able to see his history of submissions and should be give option to view any of his submitted code. If he illegally tries to access url of other user’s solution then it should be restricted.

* + - 1. Leaderboard (Tracking progress)

All registered users’ score must be visible to every user to track the progress.

* + 1. Personal Info

The user must be provided with option to change his profile and credentials(i.e. password) once that user’s session is active(i.e. after log in)

* + - 1. Update Profile

The user(Participant/setter) can edit his/her first and last name.

* + - 1. Change Password

The user(Participant/setter) can change his/her password by providing the old password once. The passwords are stored in hashed format in the database.

* 1. Performance Requirements
     1. Fairness

The system should ensure that the evaluation of a contestant submission should not be delayed indefinitely due to entries submitted later by other perticipants.

* 1. Logical Database Requirements

The system shall maintain a database of the Problems, Testcases, Users, Submissions. The system shall facilitate easy access to the data stored through the carious interfaces mentioned above.

* 1. Software System Attributes

Reliability: A high degree of reliability is necessary for this software product since this can be used for solving problems at large scale.

Security: The users of the system including the superuser, the setters, the participants needs to be authorized and authenticated to use the system. The passwords will be safely stored in the system in the hashed form. The code will be compiled and executed in an isolated environment using docker as then any malicious code submitted is going to effect only that container in which it will be running and therefore create another new container for upcoming requests.

Portability: All the dependencies will be listed out in separate requirements.txt file and the source code will either be running in the virtual environment having that dependencies or it will be running in the container having the required dependencies. This will make the source code more portable across different systems.

Accessibility: The server side rendering of frontend i.e. web pages makes it accessible to more number of users.

1. About

This Software Requirements Statement was written for the CS305 Software Engineering course project at IIT Ropar.