Software Requirements Specification

for

Automated Lab Program Evaluator

Version 1.0 approved

Prepared by:
Ashwin Joisa - 16CO104
Praveen Raj - 16CO115

NITK, Surathkal

25/02/2018

Table of Contents

1. Introduction

This Software Requirements Specification(SRS) report is prepared by Ashwin Joisa and Praveen Raj of NITK Surathkal. It mainly aims to present and describe the project Automated Lab Programme Evaluator in detail.

1.1 Purpose

The "**Automated Lab Programme Evaluator**" is a software which will be used to automate the process of evaluation of lab programs submitted by the students. This SRS captures the requirements of "**Automated Lab Programme Evaluator - Version 1.0**". This SRS is a detailed document containing all the system, functional and the non-functional requirements. This SRS also includes the revision history.

1.2 Document Conventions

In this document we have followed IEEE format and general British english conventions. In this document the requirements are listed in the decreasing order of their priorities, where every requirement has its own priority.

1.3 Intended Audience and Reading Suggestions

This document is intended for
Developers: in order to be sure they are developing the right software that fulfills requirements provided in this document.
Students, **Users**: to get familiar with the idea of the software and suggest other features that would make it even more functional.
Teachers, Teaching assistants and Testers: to get familiar with procedures involved in assigning questions and grading them and suggest more practical features.
Administrators: in order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.
Documentation writers: to know the features and in what way they have to explain. What security technologies are required, how the system will respond in each user's action.

This SRS contains the system, functional and the non-functional requirements of the software. The requirements are listed in the order, system, functional and non-functional. We suggest readers to go through this SRS in the order in which it has been written.

1.4 Product Scope

This software aims to automate the process of evaluation of lab programs written by the students. Its purpose it to make the evaluation process easier, impartial, efficient and transparent.This software intends to improve the efficiency of evaluation process by testing for more corner cases. It also reduces the burden on the teaching assistants and the teacher. This software also reduces the time taken for evaluation.

1.5 References

This SRS refers to the following external sources of information,

- The user interface style of this software has been influenced by the following article.
  Article:- User Interface and Navigation.
  Author:- Google android developers.
  Link to the article:- https://developer.android.com/guide/topics/ui/index.html

- The diagrams for this document are referenced from.
  Document:- UML Documentation
  Author:- Ashwin Joisa and Praveen Raj
  Link to the document:- **./**“UML Documentation”

2. Overall Description

2.1 Product Perspective

Automated Lab Programme Evaluator consists of a database server which contains data for Students and the problems added by Teachers. Each user’s data are divided into groups and subgroups so that they are organized in the form that serves right the user. Groups contain entries with usernames, passwords, problems solved, Submission log, maks obtained, etc. Problems database contains all the information about the problems.
In the diagram referred to below there are main components of the system, subsystem interconnections and external interfaces to help the reader understand the main idea of Automated Lab Programme Evaluator. All of them are analyzed with more detail in this document.

2.2 Product Functions

Automated lab Programme Evaluator provides the users with the following functions. Detailed use cases are given in the section 4: System Features.

- **Register**

A login screen is required for the system. In order not to get bored with lots of registration forms, an email address of the user and their password will be taken and a user profile will be created if there isn't one with that mail address already. Here usernames are Roll Numbers for Students.

- **Login**
  Registered users can login to the system after providing their usernames and passwords.
- **View a problem**
  Users that are logged into system can view a problem descriptions.
- **Submit a solution**
  Students who are logged into the system can submit a solution of the problem written in one of the permitted programming languages.
- **View results**
  Students can view the results of their submitted solution to any problem. Teachers can view the results of the whole class and can view the results of an individual too.
- **Search for a Student**
  Teachers can search for a Student in the user database (using Roll Number). The main purpose is to get his\her results and his\her submitted files.
- **Add a problem**
  Teachers can add problems to currently Exam, either their own problems or problems already in the database.
- **View submissions**
  Students can view all their submissions along with the corresponding results. Teachers can view the submissions of an individual Student.

2.3 User Classes and Characteristics

Intended user classes and their corresponding characteristics are mentioned below.

2.3.1 Students

Students are primary users of *Automated Lab Programme Evaluator*. We expect most of the "Students" to be university students, who are attending the exam.

2.3.2 Teachers/Judges

Judges will be essentially Teachers and Teaching Assistants. They have the privileges of adding questions and also view results and submissions of all the Students.

2.3.3 Administrators

Administrators are primarily responsible for maintenance of the system. They contribute minimally. They are responsible for registering new Students and new Teachers.

2.4 Operating Environment

The operating environments are listed below.

2.4.1 Operating System

This software requires to to be run on either a UNIX or Linux based Operating Systems.

2.4.2 Hardware Platform

A system with the following minimum system specifications:

- Operating System: UNIX/Linux based OS
- RAM: At least 512MB
- Storage capacity: At least 128GB
- Wifi or a network card

2.4.3 Pre-installed Softwares

The following are the required pre-installed softwares.

- GNU gcc compiler - version 5.0 and above
- GNU g++ compiler - version 5.0 and above

2.5 Design and Implementation Constraints

The following are the design and implementation constraints.

- Platform used is UNIX/Linux.
- The programming languages in which the submissions are made.
- Compatibility the pre-installed softwares (Eg: g++).

2.6 User Documentation

The user documentation would contain the following sections

1. Brief description of the software
2. Information about the Operating Environment of the software
3. Basic guide for navigating through the software.
4. References to online guides and help
5. Privacy and copyright informations

2.7 Assumptions and Dependencies

None such Assumption made or Dependencies present.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 User Interface common to all types of users

1. **Login Page**
   Features:-
   - Username
   - Password

3.1.2 User Interface specific to specific type of users

The user interface with respect to each specific type of user is described as follows.

1. **Admin :-**
   a. **Dashboard**
   Features:-
      - Add Student
      - Add Teacher

   b. **Add Teacher**
   Features:-
      - Enter Username for new Teacher account.
      - Enter Password for the new account.

   c. **Add Student**
   Features:-
      - Enter Username for new Student account.
      - Enter Password for the new account.

2. **Teacher**
   a. **Dashboard**
   Features:-
      i. Add Question
      ii. Results
      iii. Access submission logs of a particular student

   b. **Add Question**
   Features:-
      i. Add Question description
      ii. Add test cases
      iii. Specify “maximum execution time”
      iv. Specify marking scheme
      v. Done

   c. **Results**
   Features:-
      i. Results of the whole class
      ii. Result of a particular Student
      iii. Analysis after Exam

   d. **Results of whole class**
   Features:-
      i. Specify class

   e. **Results of a particular Student**
   Features:-
      i. Specify the Student

f. **Analysis after Exam**

Features:-

Gives an analysis based on number of successful submission for each question.

g. **Access submission logs of a particular student**

Features:-

i. Specify Student
ii. Access a particular submitted file.

h. **Access a particular submitted file**

Features:-

i. Specify Student
ii. Specify File name/submission number

3. **Student**

a. **Dashboard**

Features:-

i. List of question in the Exam
ii. Access Question
iii. Submit solution
iv. Results
v. Access submission logs

b. **Access Question**

Features:-

i. Specify Question number (According to the “list of question”)

c. **Submit solution**

Features:-

i. Specify the question number for which the submission is being made
ii. Specify programming language used
iii. Specify the file to be submitted

d. **Results**

Features:-

i. Marks obtained in the Exam.

e. **Access submission logs of a particular student**

Features:-

i. Access the submission log with corresponding results.
ii. Access a particular submitted file.

f. **Access a particular submitted file**

Features:-

i. Specify File name/submission number

3.2 Hardware Interfaces

The System shall be deployed on Central Server at the University premises. All the Stakeholders are supposed to log-in into this Central Server using URL to access the System. There are no specific hardware devices needed by STUDENTS or TEACHERS other than a secured access to the Central Server and a Computer to attend the Exam.

3.3 Software Interfaces

The System is self contained and no data is supposed to be shared with any third party - For instance, Students can access the questions for the exam only after logging into the server. This software requires other softwares, such as compilers, for its successful execution. The Central Server Database is implemented using MySQL database server.

3.4 Communications Interfaces

The Central Server will be available on a specific URL. Admin, Students and the Teachers will connect through a secured encrypted connection over intranet (may be LAN). This communication will be governed by TCP/IP protocol.

4. System Features

The *Automated Lab Programme Evaluator* has the following System Features and corresponding functional requirements.

4.1 System Authorization

4.1.1 Description and Priority

Description:- Every user if needs to interact with the system must be logged in with his/her credentials, hence each intended user must register to the system. Authorization guarantees the uniqueness and privacy of the user.
Priority:- High

4.1.2 Stimulus/Response Sequences

Discussed in functional requirements.

4.1.3 Functional Requirements

The following are the functional requirements of this System Feature.

REQ-1 Registration

Primary Actor : Administrator
Precondition : Connected to the database server, Logged in as ADMIN
Stakeholders and Interests : Users who want to register to the system.
Main Scenario :

a. ADMIN selects ‘Register’ option.
b. ADMIN fills in fields with the Stakeholder’s credentials. Fields like username and password.
c. ADMIN selects ‘Done’ option.
d. ADMIN is prompted about “Successful Registration”

Alternate Scenario :

2(a). Stakeholder is already registered
  a. Prompt user that he/she is registered
2(b). Stakeholder has entered a password shorter than 8 characters
  a. Ask the Stakeholder to enter longer password

`

REQ-2 Login/Logout

Primary Actor : User
Precondition : Connected the database server
Stakeholders and Interests : User who wants to enter into the system
Main Scenario :

1. User opens the login page
2. User is prompted for login and password.
3. User enters his/her username and password.
4. System does authentication
5. Main page is displayed. User can see all permitted options after login.

Alternate Scenario :

4(a). Authorization fails
  1. Prompt user that he/she typed wrong password
  2. Allow him/her to re-enter the password. Give 3 chances

4.2 Attending exam

4.2.1 Description and Priority

Description:- The Student must be able to attend an exam through this interface.
Priority:- High

4.2.2 Stimulus/Response Sequences

Discussed in Functional Requirements.

4.2.3 Functional Requirements

The following are the functional requirements of this System Feature.

REQ-1 View Problem

Primary Actor : Student
Precondition : Student has logged in
Stakeholders and Interests : Student who wants to read or solve a problem
Main Scenario:

1. User chooses a problem from the list
2. User selects “View” option and he/she is taken to problem descriptions.
3. User selects “Back” option and he/she is taken to her/his personal page

REQ-2 Submit Solution

Primary Actor : Student
Precondition : Student is logged in
Stakeholders and Interests : User or contestants wants to submit a solution to a problem
Main Scenario :

1. User goes to the page of problems
2. User chooses a problem from the list
3. User selects “Upload Solution” option, chooses a file to upload, specifies the programming language and selects upload
4. User selects “Back” button and he/she is taken to her/his personal page

Alternate Scenario :

3(a): User has not specified any programming language

1. Prompt user that he/she has not chosen any programming language
2. Take user to solution submission page

REQ-3 View Submissions

Primary Actor : Student
Precondition : Student has logged in
Stakeholders and Interests : Students who wants to see his/her previous submissions to a problem
Main Scenario :

1. Student selects "Problems" option.
2. System provides all available problems
3. Student selects "View Submissions" option for a particular problem in the list.
4. System provides all solution submissions to that problem
5. Administrator selects "View" option in order to see submitted solution

4.3 Evaluation

4.3.1 Description and Priority

Description:- The major aim of this software is to evaluate the lab programmes submitted by the students. The output of this feature is the results after evaluation.
Priority:- Medium to High

4.3.2 Stimulus/Response Sequences

Discussed in Functional Requirements.

4.3.3 Functional Requirements

The following are the functional requirements of this System Feature.

REQ-1 Add problem

Primary Actor : Teacher
Precondition : Teacher is logged in
Stakeholders and Interests : Teacher who wants to add problems for the exam
Main Scenario :

1. Teacher selects "Add own problem" option.
2. Teacher specifies the required fields
    1. Restricted programming languages
    2. Maximum execution time
    3. Teacher uploads the problem description
    4. Teacher uploads the test cases
    5. Teacher selects "Done" option

Alternate Scenario :

1(a) : Teacher presses “Add problem from site collection”

1. Teacher selects a problem from problem list already in the database
2. Teacher selects “Add” option.

REQ-2 View Results

Scenario #1:

Primary Actor : Teacher
Precondition : Teacher is logged in
Stakeholders and Interests : Teacher who wants to access results
Main Scenario :

1. Teacher selects “Results” option.
2. Teacher specifies the class whose result is required.
3. The results of the whole class is displayed.
4. Teacher can download it.
5. Teacher selects “Done” option.
6. Teacher is returned to his/her main page.

Alternate Scenario :

2(a) : Teacher specifies the Student whose result is to be accessed.

1. The results of the specified student is displayed along with his/her submission logs.
2. Teacher can download the result.

2(b) : Teacher specifies the Student whose result is to be accessed.

1. The results of the specified student is displayed along with his/her submission logs.
2. Teacher selects a file from the submission log to view it.
3. The file selected is displayed.
4. Teacher selects “Done” option.

Scenario #2:

Primary Actor : Teacher
Precondition : Teacher is logged in
Stakeholders and Interests : Teacher who wants to access results
Main Scenario :

1. Teacher selects “Results” option.
2. Teacher specifies a Question whose result analysis is required.
3. The results analysis statistics is of that question w.r.t the whole class is displayed.
4. Teacher can download it.
5. Teacher selects “Done” option.
6. Teacher is returned to his/her main page

Scenario #3:
Primary Actor : Student
Precondition : Student is logged in
Stakeholders and Interests : Student who wants to access results
Main Scenario :

1. Student selects “Results” option.
2. Student specifies the Question whose result need to be accessed
3. The results of that question is displayed.
4. Student selects “Done” option.
5. Student is returned to his/her main page.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The following are the performance requirements of the system.

1. **The system should support at least 5 concurrent user requests**:- During an exam, since many Students would be submitting solutions and querying results and questions, the server must be able to handle atleast 5 concurrent requests. This provides a general sense of reliability when the system under load.
2. **The system must be able to recognise erroneous programmes submitted by students:-** There might be programmes that the students submit which might run into infinite loop, hence this system must recognise them. This is essential for smooth running of the software.
3. **The system must be able to complete the testing process for a particular test file in at most the time allotted to that question:-** This ensures faster result analysis and gives a sense of reliability of the system.

5.2 Safety Requirements

No such requirements are necessary.

5.3 Security Requirements

The following are the Security Requirements.

1. **An Authorisation mechanism :-** This is the mechanism, that is required to authorise a user as either an ADMIN or a STUDENT or as a TEACHER. This is important since each of these types of users have their own special privileges which if runs into the other user, the result would be disastrous.
2. **A Login mechanism :-** This mechanism is necessary since each of the user of the system is required to be logged in before interacting with the system interface. This is because, an user who is not logged in is not yet recognised as either ADMIN, STUDENT or TEACHER.

5.4 Software Quality Attributes

In this section software attributes are specified in order to verify them objectively.

1. **Usability**
As far as possible, the system should provide a simple, responsive interface.
Although Automated LAb Programme Evaluator is composed of diverse systems, applications, and processes, the underlying architecture should be transparent to the administrators. The system should be configurable from a single source at a central administrative position, and should provide a central, easy-to-use interface that will allow administrators to configure the user interface and features in a way that reduces page clutter.
A system will be considered to meet this requirement if it has a single administrative interface rather than individual links for editing each page.

2. **Reliability**
The system must be backed up on a configurable schedule.
Back-up requirements will need to be determined, based on individual components of the system. The system should be backed up with a frequency that ensures system and all data is protected. Since the updates and changes will be done to the database via web-interface, it should be backed up on a nightly basis, with options for weekly, as well as off-site backup when necessary.
The system must have the ability and capacity to restore backup data within five hours so that the system is never offline for an inordinate period of time.

3. **Availability**
The system should be available 24 hours a day, 7 days a week.
This statement provides a general sense of system availability. It is not intended to demand the system maintain reliability, or to require the system to be highly available. It should not exclude scheduled downtime. It is only intended to convey the expectation that our users should have access to the system during organized contest times which makes this attribute number one priority;

4. **Maintainability**
The system must be maintainable without substantial modification
Due to limited number of administrators and support staff, it is important that the system be mostly self-sustaining. This will reduce the number of hours spent maintaining the system and simplify maintenance tasks.

5. **Performance**
The system should support at least 5 concurrent users.
This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of actors be able to access the system at the same time.

6. **Security**
   The system must comply with the permission roles
   Security is the most important attribute of the system. System should not allow unauthorized accesses. It should not mix the roles of the users.

5.5 Business Rules

The following the are the roles of each individuals, based on their authorisation status.

1. **ADMIN :-**
   a. Register an user as STUDENT or as TEACHER.

2. **TEACHER :-**
   a. Add Questions and corresponding Test Cases.
   b. Query for a particular submitted file of a particular STUDENT.
   c. Query of results of either the whole class or of an individual STUDENT.
   d. Query for an analysis of the Exam based on no of successful submissions for each question.

3. **STUDENT :-**
   a. Query for the Question that he/she wants to solve.
   b. Submit the solution for a particular question.
   c. Request for his/her results.
   d. Query for his/her Submission logs.
   e. Access his/her previously submitted solution for a particular question.

6. Other Requirements

1. **Database Servers Requirement**
   MySQL Database server is required for a centralized storage system. We select MySQL because MySQL is free software and according to researchers it has most of the abilities of a DBMS should have. Additionally it is easy to host, more powerful and useful.

2. **Web Server Requirement**
   We need a fast and reliable server. This machine will show high performance. For web server we will use a dual-core 2.6 Ghz, 2 Gb DDR-2 ram with at least 800 Mhz frequency. Its capacity is not very important. Standard capacity will be sufficient. This server must have a very high bandwidth or maybe more servers. Any IBM based computer will be useful, its trade is not important. Any dedicated web server is seeming as mandatory. Moreover it is not mandatory yet desirable if it has very high speed connection

3. **Developers Requirements**

Project developers must have at least 1.6 Ghz Intel CPU, 512 Mb DDR2 Ram, with Unix/Linux Operating system. And 20Gb hard-drive will be sufficient.

Appendix A: Glossary

- IEEE format for SRS : refer https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc
- UNIX base OS : eg:- Darwin, Mac OS X.
- Linux based OS : eg:- Ubuntu, Linux mint, Fedora.
- Database servers : eg:- Oracle, DB2, Informix, PostgreSQL, etc.
- MySQL database servers :- refer https://www.mysql.com/

Appendix B: Analysis Models

The following are the analysis models for this software

1. Use Case Diagram

2. Class Diagram

3. Activity Diagram

4. Sequence and Collaboration Diagram

5. State Machine Diagram

6. Component Diagram

7. Deployment Diagram

Each of these diagrams are mentioned in the UML documentation document. The links for these diagram have been hyperlinked with the names mentioned here (i.e click on the diagram name above and it will take you to the diagram location).