

CLOUD BASED ONLINE PROGRAMMING PORTAL

AN INDUSTRIAL INTERNSHIP REPORT

submitted by

SIDDHANT VERMA

13BCE1146

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



AUGUST,2016



School of Computing Sciences and Engineering

DECLARATION BY THE CANDIDATE

I hereby declare that the Industrial Internship Report entitled “**CLOUD BASED ONLINE PROGRAMMING PORTAL**” submitted by me to VIT University, Chennai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide industrial training undertaken by me under the supervision of **Dr. Nayeemulla Khan**. I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Signature of the Candidate

Date:

TABLE OF CONTENTS

Chapter	Title	Page
	Declaration	i
	Certificate	ii
	Table of contents	iii
	Bonafide Certificate	Iv
	Acknowledgement	V
	Abstract	Vi
1.	Introduction	
	1.1 About the system.	1
	1.2 About the company.	2
2.	Overview of the system	3
3.	Design	9
	3.1 Overview	9
	3.2 Client – Server architecture.	9
	3.3 Virtualization on cloud.	9
	3.4 Database and file systems.	10
	3.2 UML diagrams.	12
	3.2.1 Use Case diagram.	12
	3.2.2 Flow Charts	13
	3.2.3 Sequence diagram	16
4.	Implementation	17
	4.1 Student module	17
	4.2 Admin module	23
	4.3 Faculty module	29
	4.4 Java Server	31
5.	Conclusion and future work	32
	References	33



School of Computing Sciences and Engineering

BONAFIDE CERTIFICATE

This is to certify that the Industrial Internship Report entitled “**CLOUD BASED ONLINE PROGRAMMING PORTAL**” submitted by **SIDDHANT VERMA (13BCE1146)** to VIT University, Chennai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in**

Computer Science and Engineering is a record of bonafide internship undertaken by him/her fulfills the requirements as per the regulations of this institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Dr. Ganesan
Programme Chair (B.Tech CSE)

Date:

Examiner (s) Signature

1.

2.

ACKNOWLEDGEMENT

Internships are the best way to get a practical knowledge of the theoretical concepts learnt during our Engineering classes. And during this phase of extensive learning, there are a lot people who have helped me during this internship. I would like to acknowledge all these people here in this report.

Firstly, I would like to thank, **Dr. Nayemulla Khan**, Dean, Research, VITCC and also the internship coordinator. He guided us during the entire internship. His helpful nature and guidance helped me to learn a lot of new things, which I would otherwise be deprived of. His experience in the field of Cloud Computing was really inspiring. Secondly, I would like to thank, **Mrs. Janaki Meena**, Professor, VITCC, who gave me different sort of problem statements to work on and guided me throughout the internship. Also, **Mr. Sathis Kumar**, professor, VITCC, guided me in the field of software standards. He helped me explore this field of system security.

Siddhant Verma.

ABSTRACT

In the last decade, problem solving and programming have emerged as skills that every student must possess. It is imperative for universities to help students learn these skills, and apply them effectively. This can be achieved by making students to practice problems related to problem solving. This will help them to grasp the theory concepts effectively, understand the different approaches of solving a particular problem, and analyze the applications of various concepts.

Our project aims at developing an online programming portal for the students of VIT Chennai. The portal allows professors to post problems which test the programming skills of students. The students in turn can solve the problems in C, C++, Python, or Java, depending upon the language specified by the problem setter (Or, students can choose any language if it is not specified by the problem setter). The portal also enables professors to evaluate the problems posted in it. To compile programs in different languages, a Java server is being used.

CHAPTER 1: INTRODUCTION

1.1 About the System

The programming portal consists of a web application which acts as an interface to the students and teachers. The web application is divided into three modules, and they are as follows –

1. Student Module –

This module allows a student to login to his/her account, view problems posted by the admin, and solve them.

2. Faculty Module –

This module allows a teacher to login to his/her account, view the submissions posted by students, and evaluate them.

3. Admin Module -

This module allows an admin to login to his/her account, schedule new tests for the students, post new problems, view student and faculty details, and view the submissions posted by students.

Since the portal supports execution of code in C, C++, Java and Python, a Java server is used for the compilation process. The java server is required to use the compilers installed in the underlying OS, to execute the code, depending upon the language chosen by the student. Whenever a program needs to be executed, the web application, which is written in PHP, communicates with the Java server through a socket. The java server then executes the program and sends the output to the web application.

Thus, in a nutshell, the process can be summarized as follows –

- Admin/Faculty posts question.
- Student views the problem, and attempts it.
- Faculty views the solution of a particular student to a particular problem, and evaluates it.

This depicts the system in a highly abstract manner. A more detailed explanation of the design and working of the system can be found under the “design” and “implementation” sections.

1.2 About the Company

VIT University was established under Section 3 of the University Grants Commission (UGC) Act, 1956, and was founded in 1984 as a self-financing institution called the Vellore Engineering College. The Union Ministry of Human Resources Development conferred University status on Vellore Engineering College in 2001. The University is headed by its founder and Chancellor, Dr. G. Viswanathan, a former Parliamentarian and Minister in the Tamil Nadu Government. In recognition of his service to India in offering world class education, he was conferred an honorary doctorate by the West Virginia University, USA. Sankar Viswanathan, Sekar Viswanathan and G.V. Selvam are the Vice-Presidents; Dr. Anand A. Samuel is the Vice-Chancellor and Dr. V. Raju and Dr. S. Narayanan are the Pro-Vice-Chancellors.

VIT University was established with the aim of providing quality higher education on par with international standards. It persistently seeks and adopts innovative methods to improve the quality of higher education on a consistent basis. The campus has a cosmopolitan atmosphere with students from all corners of the globe. Experienced and learned teachers are strongly encouraged to nurture the students. The global standards set at VIT in the field of teaching and research spur us on in our relentless pursuit of excellence. In fact, it has become a way of life for us. The highly motivated youngsters on the campus are a constant source of pride. VIT's Memoranda of Understanding with various international universities are one of its major strengths. They provide for an exchange of students and faculty and encourage joint research projects for the mutual benefit of these universities. Many of VIT's students, who pursue their research projects in foreign universities, bring high quality to their work and esteem to India and have done the university proud. The University has two campuses – One in Vellore, and the other in Chennai.

CHAPTER 2: OVERVIEW OF THE SYSTEM

VIT University has been using SkillRack for academic purposes for some time and wanted an in-house version of the same, hence the ultimate objective of the project was to emulate Skillrack with additional features as recommended by the guides and supervisors of the same.

FRONT END:

HTML & CSS:

HTML or HyperText Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . A web browser can read HTML files and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

Basic HTML page structure:

```
<html>
  <head>
    <title>Welcome to HTML</title>
  </head>
  <body>
    <h1>This is heading 1.</h1>
    <p>And this is a paragraph</p>
    <div>
      
      <p>This is a paragraph in a Div element.</p>
    </div>
  </body>
</html>
```

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain Extensible Markup Language (XML), Scalable Vector Graphics (SVG) and XML User Interface Language (XUL). CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

JAVA SCRIPT:

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions (first class functions can be passed as arguments to other functions, return as the values from other functions, and assign to variables or storing them in data structures). JavaScript is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

A Hello world program in JavaScript:

```
<html>
  <head>
    <title>Hello World Program in JavaScript </title>
  </head>
  <body>
    <h2>Hello World</h2>
    <script language="JavaScript">
      document.write("Hello World from Javascript");
    </script>
  </body>
</html>
```

JQUERY:

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. Used by over 60% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today. jQuery is free, open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.

jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

PHP:

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page – for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

The following Hello world program is written in PHP code embedded in an HTML document:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

MODULE 1:

This module consists of the part that facilitates the work of the admin. The landing page consists of the code tests that have been scheduled and they are available chronologically according to the test id. The admin module has features like scheduling a test, creating problems for the test, candidate management, report generation, resetting a password, creating a email group.

On the landing page with every scheduled test, the admin also has the option to change the details of the test or view its status. The next option available for the admin is to schedule a code test which involves selecting the batch, the class numbers, begin key (if needed for the test) and then a test id is automatically generated.

Next, the admin can create a problem. Creating a problem involves entering the details like the title of the problem, the description, the pre-code, the post-code, the duration for which the code must run hidden and visible test cases and their outputs.

The admin also has the ability to create an email group in the event of a few people missing the test so that a separate test can be conducted for them. The password reset option gives the admin the power to change password for students, faculties and fellow administrators (if any). Report generation enables the admin to generate a report of a particular test that has been undertaken. Candidate management takes registration number as the input and gives the details of a particular candidate along with the option to edit it.

MODULE 2:

Module 2 involves the faculty end of the software and includes the features for authentication and more importantly the feature to evaluate the test of a student. The faculty can see the various tests scheduled in their respective classes and when it clicks on evaluate portion, the faculty is shown the list of students who have submitted the test and is also shown the code written by them and the status of both the hidden and visible test cases. In case all the test cases passed the faculty is just shown the message that the test cases have passed but if there are a few that worked and a few that did not then the faculty is shown the complete details. The faculty can then go ahead and evaluate the test as per the criteria.

The faculty also has the ability to change the password for a student.

MODULE 3:

Module 3 involves the student end of the system. The student after logging in sees all the tests scheduled in his/her classes and after clicking on a particular test the student sees all the problems in the test. When the student goes ahead and clicks on a particular problem, the details of the problem are shown along with the code editor to type the code. The code editor has been built using an open source module and has been extensively customised to suit the needs of the project. The student has the option to choose between the languages C, C++, Python and Java in case the language has not been frozen.

The student further has two options, first to compile and run and second to submit. The first option checks for the users code against the visible test cases while the submit option checks the code against hidden test cases. On successful compilation and submission a success message is shown. Else if the output did not match for one of the visible test cases, then the suitable error is shown else the visible test cases pass but there is a problem in one of the hidden test cases then only an error message is shown and not the complete details of the error.

BACKEND:

My SQL:

MySQL is an open source relational database management system. MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

MySQL Workbench is the official integrated environment for MySQL. It was developed by MySQL AB, and enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

For security purposes we used parameterized queries while coding the backend. A parameterized query (also known as a prepared statement) is a means of pre-compiling a SQL statement so that all you need to supply are the "parameters" (think "variables") that need to be inserted into the statement for it to be executed. It's commonly used as a means of preventing SQL injection attacks.

Amazon Web Services (AWS):

Amazon Web Services, a subsidiary of Amazon.com offers a suite of cloud-computing services that make up an on-demand computing platform. These services operate from 13 geographical regions across the world. The most central and best-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3". As of 2016 AWS has more than 70 services, spanning a wide range, including compute, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools and tools for the Internet of things. Amazon markets AWS as a service to provide large computing capacity quicker and cheaper than a client company building an actual physical server farm. It also accounts for the basic cloud facilities like auto scaling and load balancing and was used as an Infrastructure as a service by setting up a VM on the cloud and then having a separate thread for each user.

CHAPTER 3: DESIGN

3.1 OVERVIEW:

The front end consists of three portals – admin, faculty, and student logins. Admin can log on to his portal and create tests and add problems to the test. The test can be scheduled for a class or more than one class. Moreover, admin has access to all the student details including marks and personal details. In faculty login, professors can see the submissions of the students and evaluate the submissions. They can access the details of the students registered under them. In the student login, student can login and see all the tests scheduled for them and can take those tests.

3.2 CLIENT –SERVER ARCHITECTURE:

In the backend, PHP and Java server is used. The java server is hoisted on a Windows Server virtual machine running on Amazon Web Services cloud. PHP code acts as a client and opens socket for each user. The server keeps running in the back and keeps on waiting for the connection request from PHP client. When the compile or test button is clicked, a connection request is placed by the client. As soon as it receives a connection request from the client, a thread of server is created and a connection is established between PHP client and Java server. The code is transferred to the java server. The java server writes the code in a file. Java server also receives the information about the language used in the code from the client. Java server will create a batch file to compile the code. Upon the creation of the batch file, the server invokes a system call to run that batch file to compile the code. If the compilation process is successful, the output is written to a text file or if the compilation fails, error log is written in a text file. This text is returned to PHP client using the same socket.

In case of successful compilation, a batch file is created to execute the code. This batch file is invoked and the code is validated against the test cases. The output of the code is written into a text file and compared with the expected output. If the output matches, success message is sent to client else failure message is sent to the client. The PHP client upon receiving the message from the server displays the message.

When the submit button is clicked, the code is validated against the hidden test cases . In this case, the test cases are not shown to the user. If all hidden test cases pass, success message is sent to the client else failure message is returned to the client.

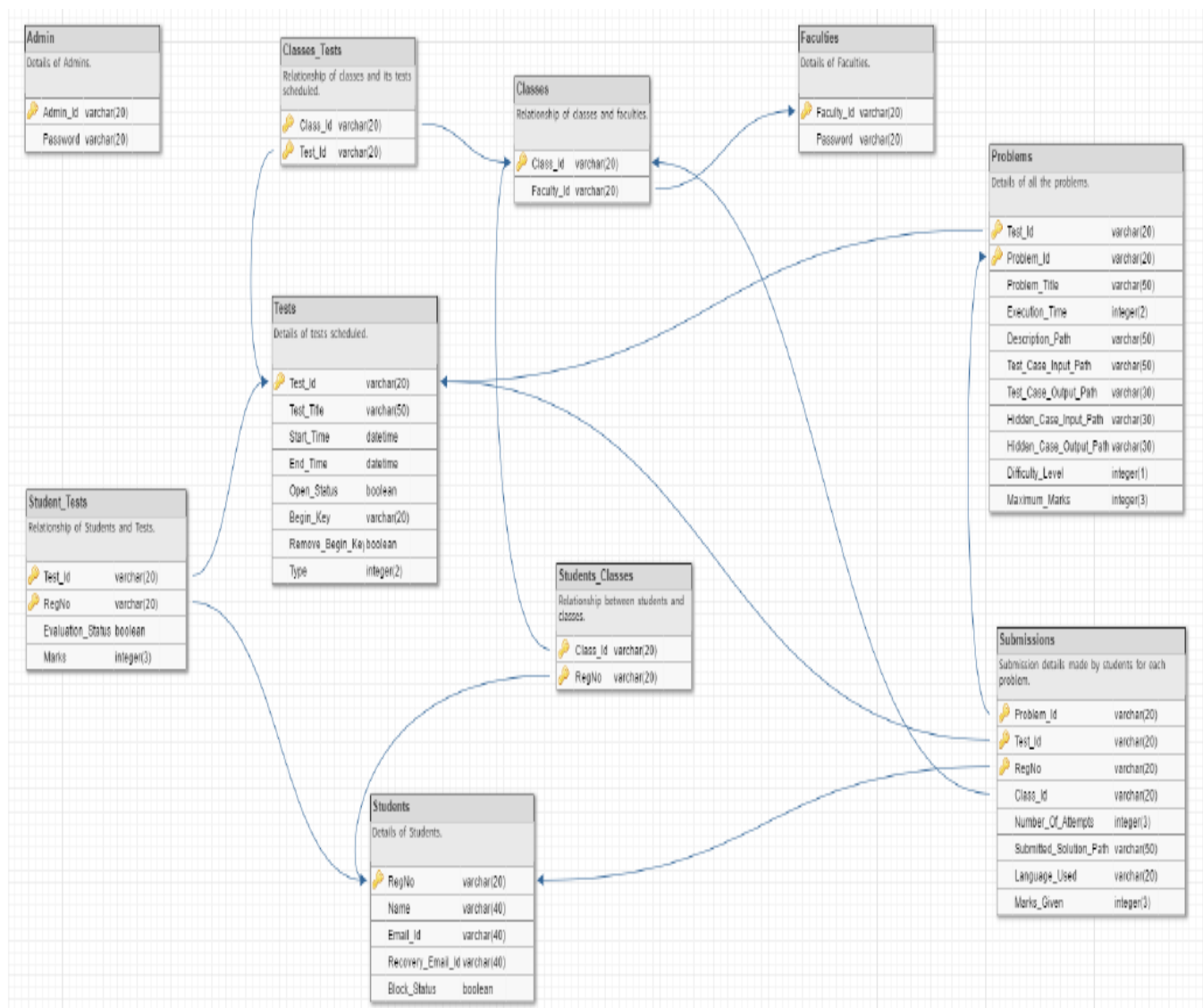
3.3 Virtualization on cloud:

The portal runs on a Windows Server virtual machine hosted on Amazon Web Services cloud. The virtual machine runs all the backend processes required to run our system. These processes include XAMPP server for running PHP script, a Java server and all the compiler services. This virtual machine is accessible on internet using its public IP address. We can access this virtual machine using remote desktop client using RDP protocol. The VM can listen to incoming HTTP request from the client. The VM helps to make the portal scalable and helps in balancing the load.

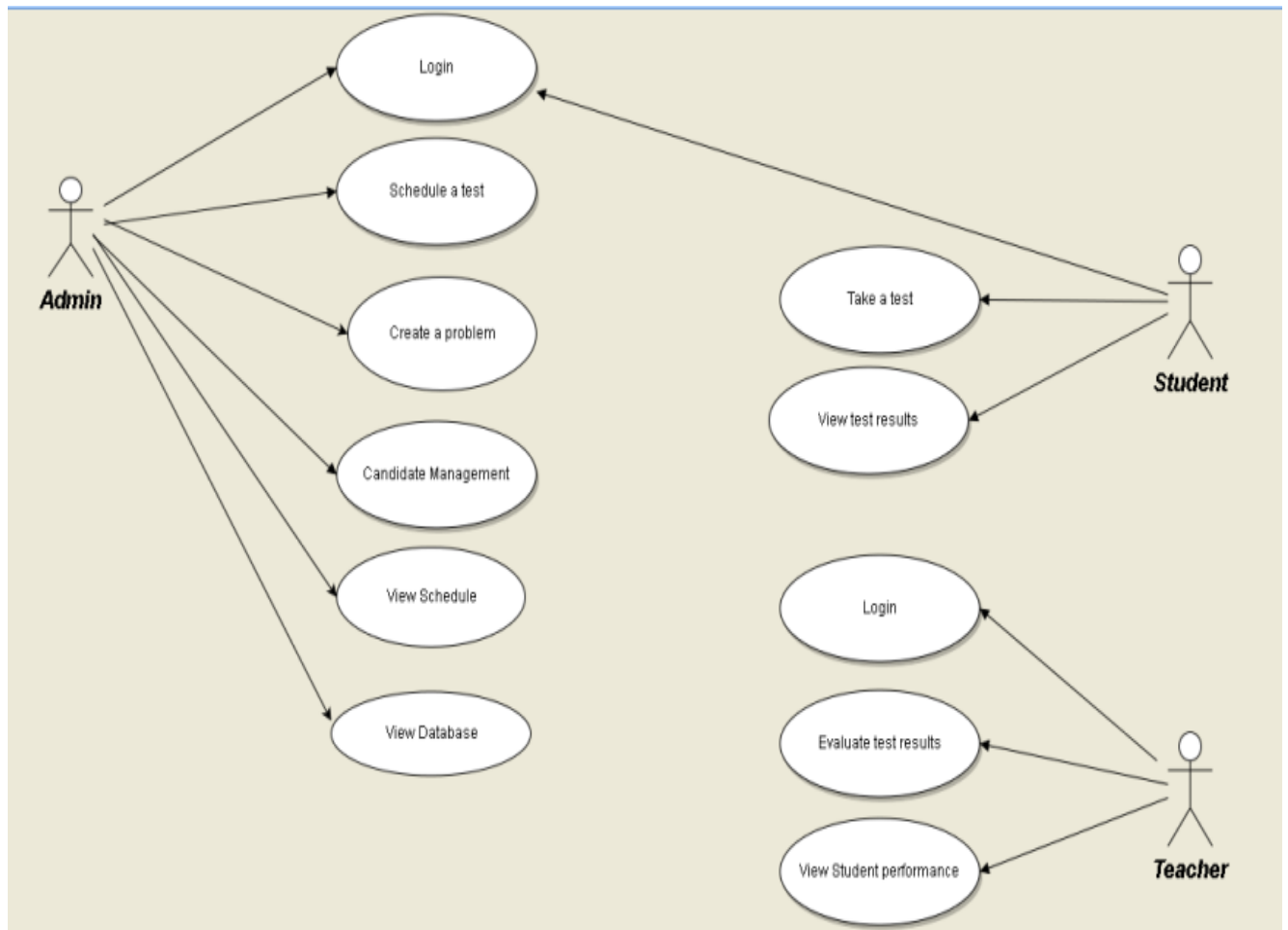
3.4 Database and File Systems:

This project uses amalgamation of two systems, namely database and file systems for data handling. MySQL database has been used for storing the data in relational format. Most of the data in our system is stored in MySQL database. But, there are certain fields like program descriptions, codes, test cases whose size cannot be pre-determined. One of the options was to store these fields in database by keeping the size of these attributes at very high value. But, this approach is highly inefficient and leads to wastage of memory. So, we have stored such attributes in a file system. These attributes are stored in text files and the paths of these text files are stored in the database. Therefore, whenever the server wants to access any of these fields, it reads the path of the corresponding file in database and accesses the file.

Database schema:

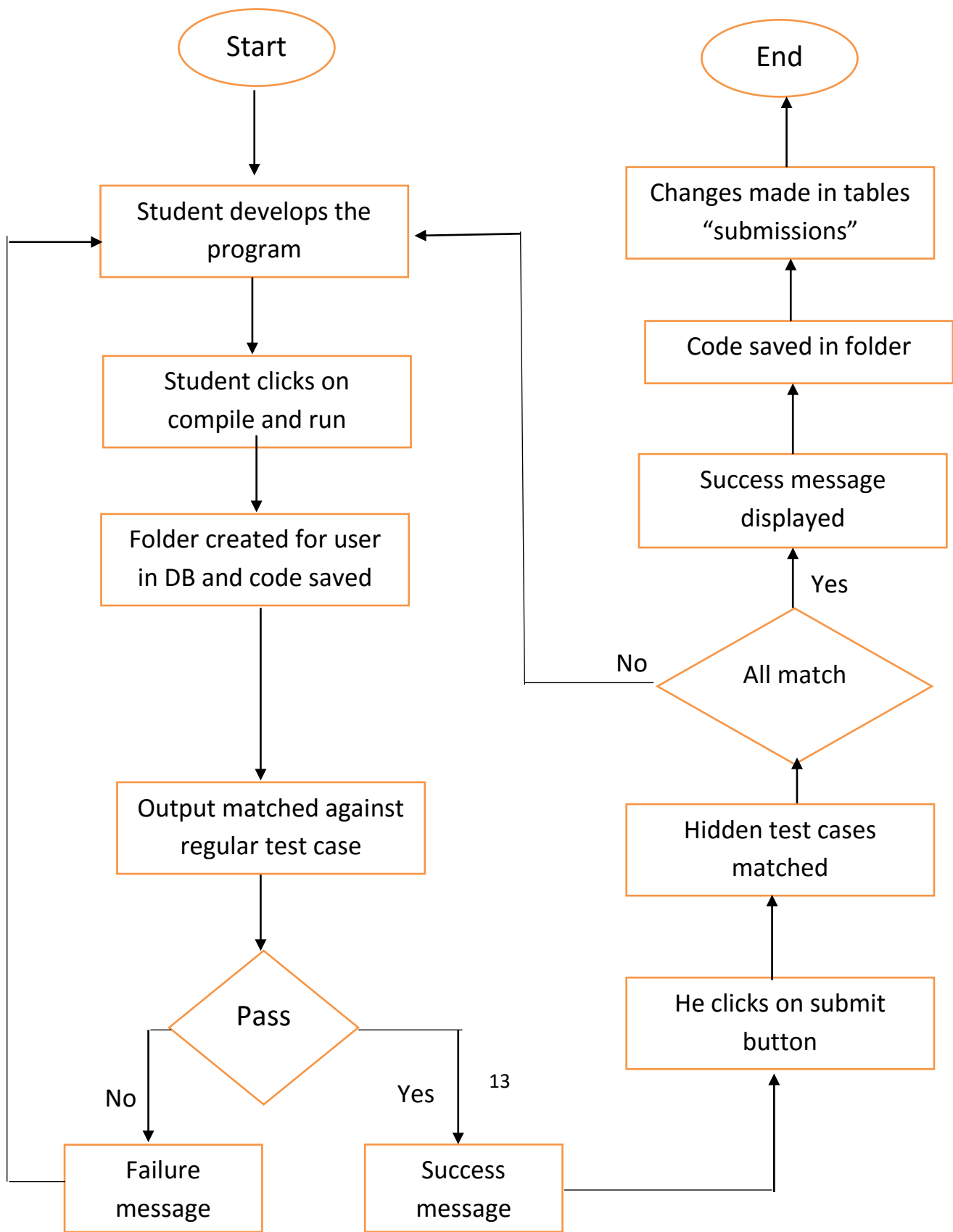


3.5 Use Case diagram:

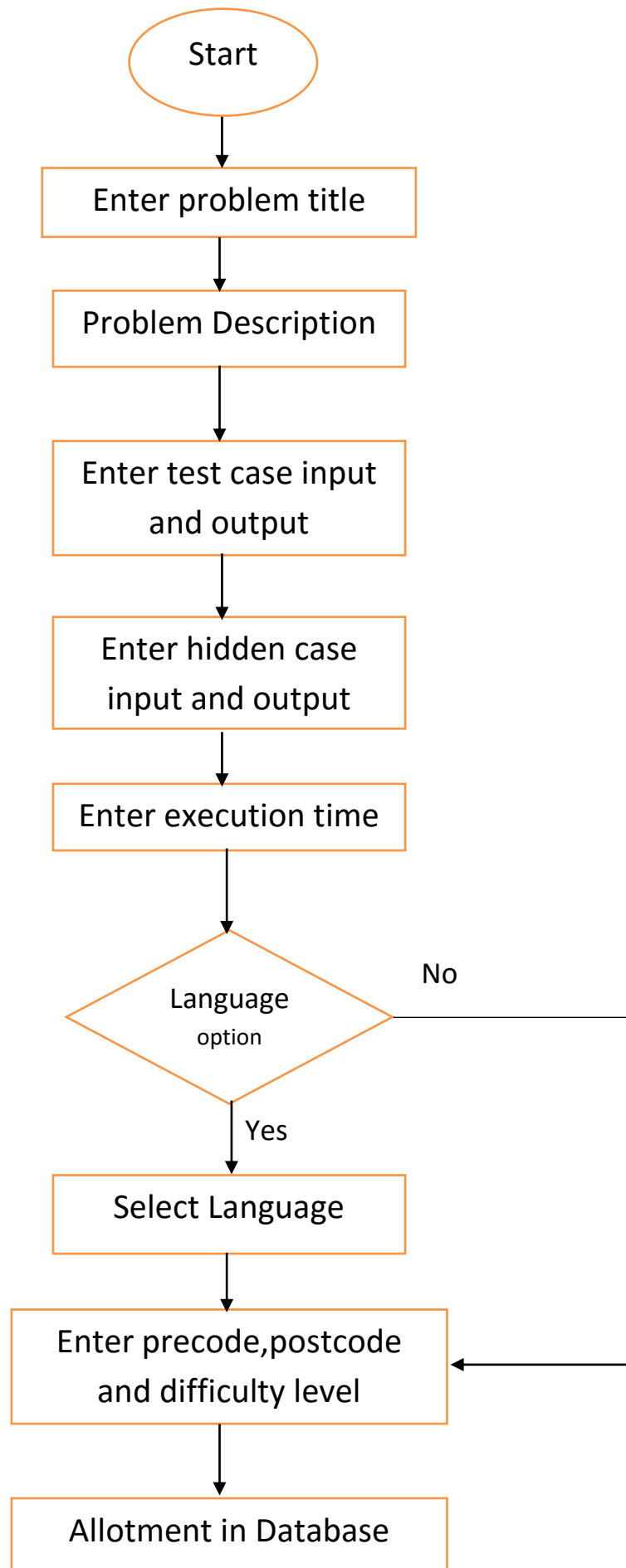


Submission of solution

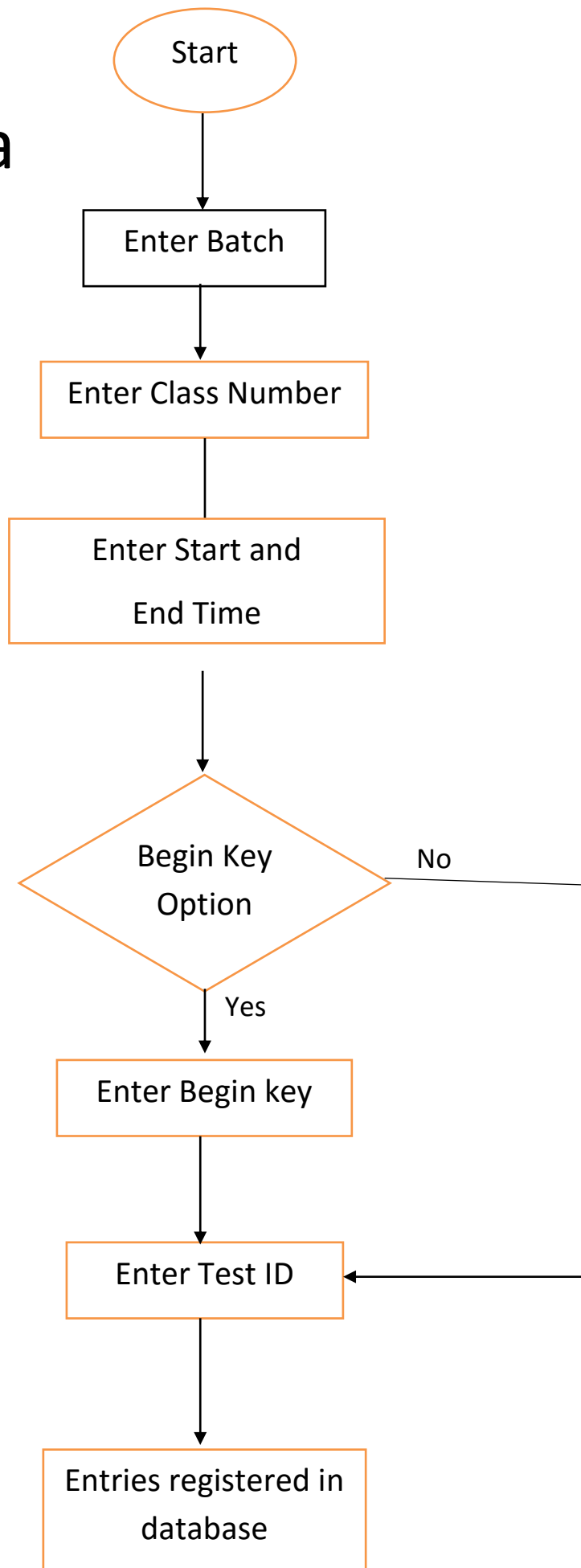
3.6 Flow charts:



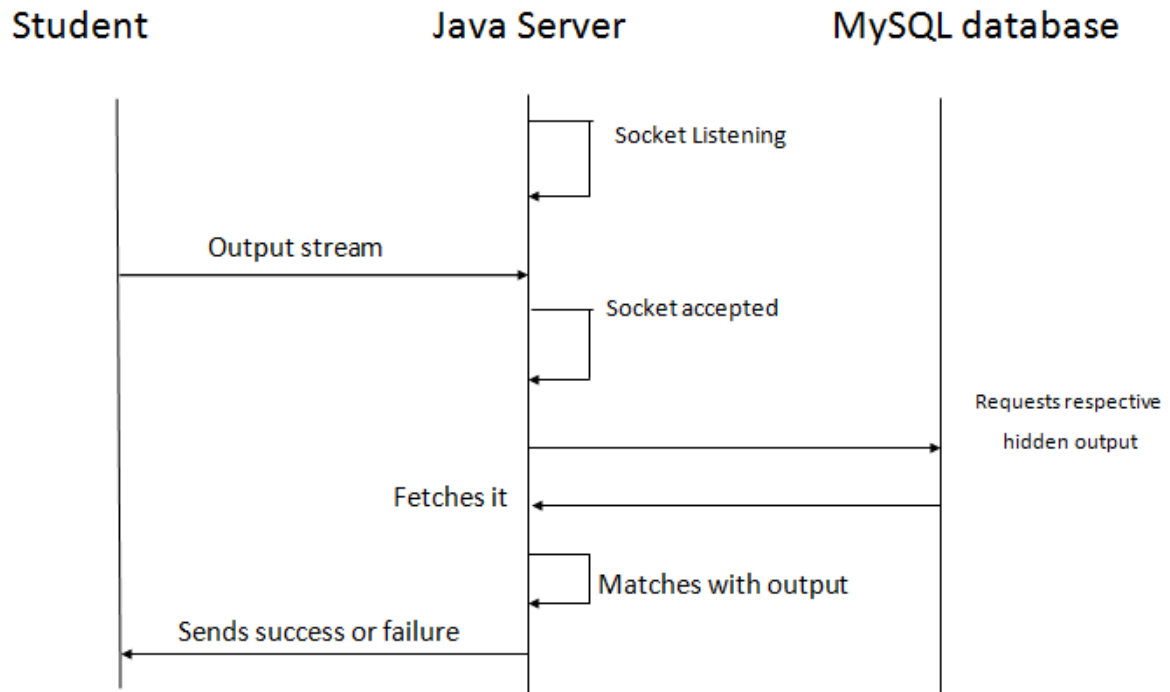
Create a
problem



Schedule a test

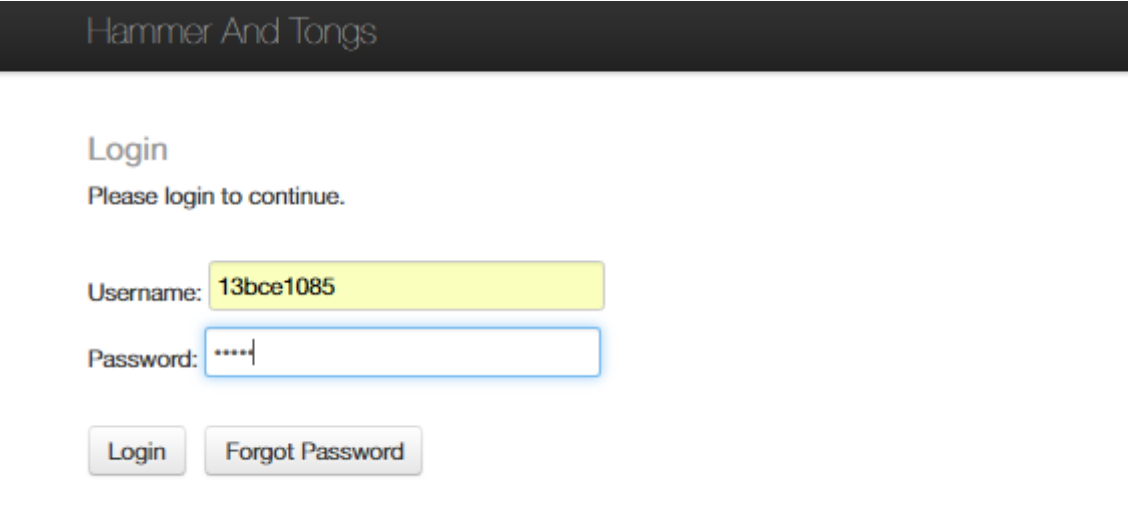


Sequence diagram:



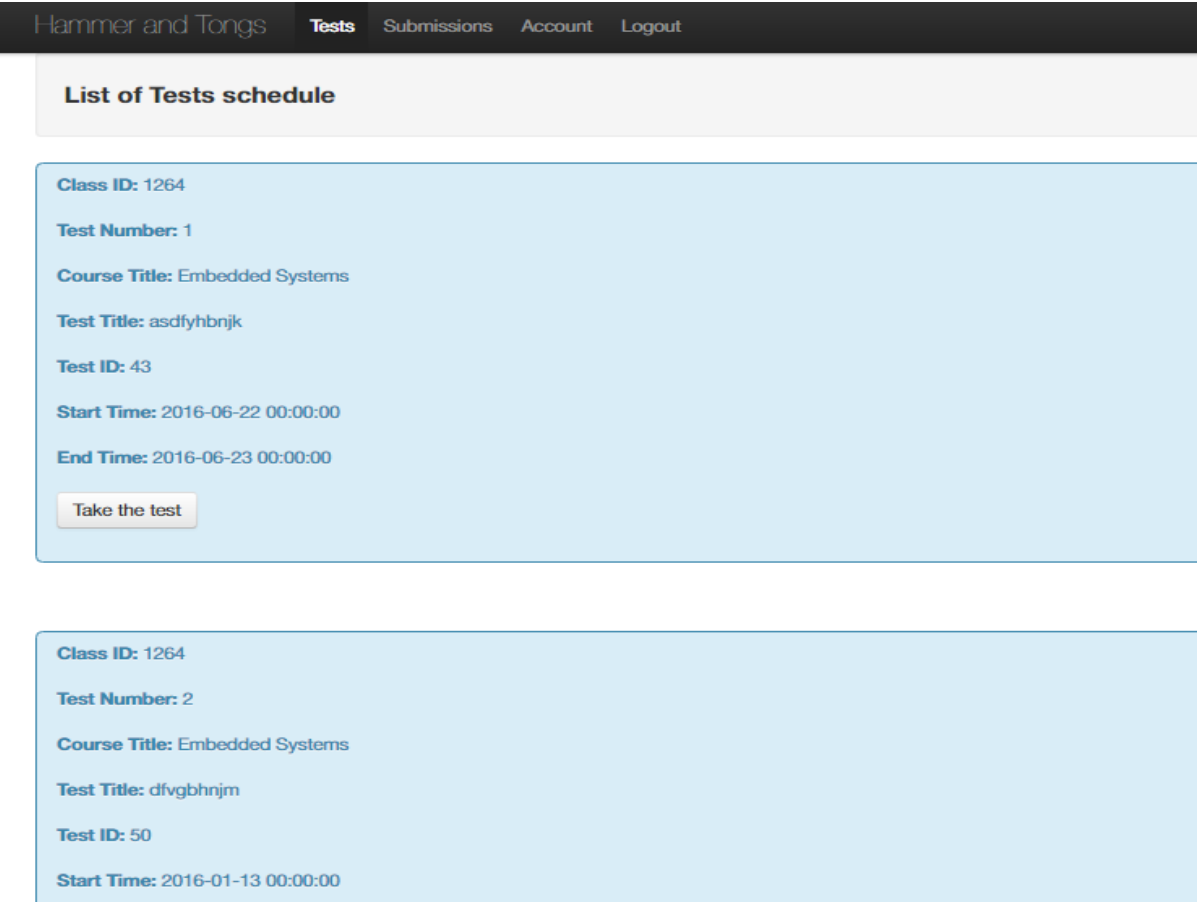
CHAPTER 4: IMPLEMENTATION

4.1 Student Login



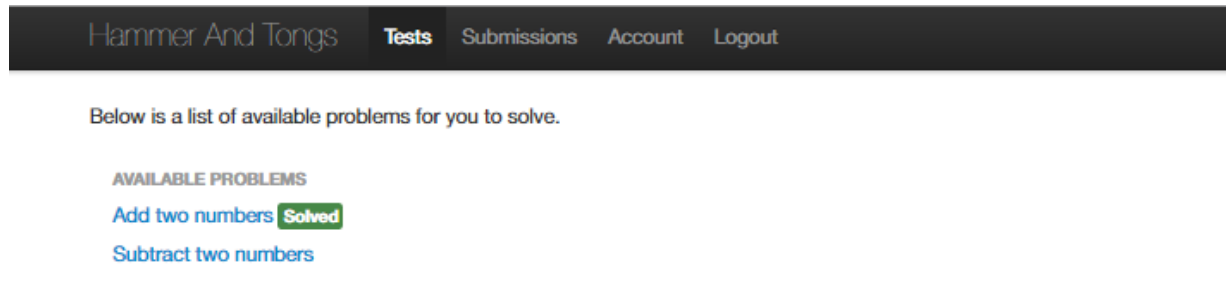
The screenshot shows the login interface for 'Hammer And Tongs'. At the top, the name 'Hammer And Tongs' is displayed in a light blue font on a dark blue background. Below this, the word 'Login' is centered, followed by the instruction 'Please login to continue.' There are two input fields: 'Username:' with the value '13bce1085' and 'Password:' with masked characters '*****'. Below the password field are two buttons: 'Login' and 'Forgot Password'.

A student can login to his/her account by entering his/her registration number, and the chosen password. Once the student logs in, he/she can view the tests that have been scheduled.



The screenshot shows the 'List of Tests schedule' page. At the top, there is a navigation bar with 'Hammer and Tongs' and links for 'Tests', 'Submissions', 'Account', and 'Logout'. The main heading is 'List of Tests schedule'. Below this, there are two test entries, each in a light blue box. The first entry shows 'Class ID: 1264', 'Test Number: 1', 'Course Title: Embedded Systems', 'Test Title: asdfyhbnjk', 'Test ID: 43', 'Start Time: 2016-06-22 00:00:00', 'End Time: 2016-06-23 00:00:00', and a 'Take the test' button. The second entry shows 'Class ID: 1264', 'Test Number: 2', 'Course Title: Embedded Systems', 'Test Title: dfgbhnjnm', 'Test ID: 50', and 'Start Time: 2016-01-13 00:00:00'.

The student can now view all the tests scheduled for him/her. Relevant details regarding a particular test, such as the Course Title, the test start time, and the test end time, are also displayed. Now, once the student clicks on the “Take the Test” button, the problems available under the particular test are displayed –



Under a section titled “AVAILABLE PROBLEMS”, the student can view all the problems which belong to the particular test. A student can select any one of the problems, and solve it. He/she can also view the status of a particular problem (Attempted the problem, Solved the problem). After a problem is selected, the next page shows the title, and the description of the problem.

Below is a list of available problems for you to solve.

AVAILABLE PROBLEMS

[Add two numbers](#) **Solved****Subtract two numbers**

Subtract two numbers

Given two numbers, x and y , print the difference of the two numbers.

Input -

Two numbers, x and y .

Output -

Difference between x and y .

Sample Input -

5 4

Sample Output -

1

[Solve](#)**0**

have solved the problem.

This page provides a detailed description of the selected problem. This description consists of the problem statement, the expected input, the expected output, the sample input and the sample output. If the student wishes to solve this particular problem, he/she has to click the "SOLVE" button, which will direct him/her to a code editor page.

Language: C++ ▾

Type your program below:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int i,j;
6     cin>>i>>j;
7     cout<<i-j;
8 }
9
```

Compile and Test

Submit

You are allowed to use any of the following languages: C, C++, Java, Python

Congratulations! The sample testcases have passed.

The code editor in this page allows the student to write a program in 4 languages, namely, C, C++, Python and Java. The drop down option above the code editor allows the student to select a language. There are two buttons shown here –

1. **Compile and Test** - When the Compile and Test button is clicked, the sample input is fed to the program. The output corresponding to this input is displayed to the student, along with the expected output. This allows the student to view the cases for which his/her code has passed, and the cases for which it has failed. Also, since the expected output is also shown, it becomes easier for the student to find the possible mistake(s) in the program.
2. **Submit** – When the submit button is clicked, the hidden test cases are fed to the program. These test cases are not shown to the student. So, even if one test case fails, a message is shown to the student which indicates that one or more test cases have failed. But the test cases which have passed, or failed, are not shown to the student.

The sample problem shown in the submission involves subtraction of two numbers. The following snapshots depict the different messages displayed to the student, depending upon the program submitted, and the button that he/she has clicked.

Language: Java ▾

Type your program below:

```
1 import java.util.*;
2 public class solve
3 {
4     public static void main(String args[])
5     {
6         Scanner scanner = new Scanner(System.in);
7         int a = scanner.nextInt();
8         int b = scanner.nextInt();
9         System.out.println(a+b);
10    }
11 }
12
13
14
15
16
```

Compile and Test

Submit

You are allowed to use any of the following languages: C, C++, Java, Python

1. The logic of the code written here generates the wrong output. Thus, the output generated by both “Compile and Test”, and “Submit” are as follows –

a. Compile and Test

Compile and Test

Submit

You are allowed to use any of the following languages: C, C++, Java, Python

Your program output did not match the solution for the problem. Please check your program and try again.

Input -

5

4

Expected -

1

Received -

9

Failed

b. Submit

Compile and Test

Submit

You are allowed to use any of the following languages: C, C++, Java, Python

Your program output did not match the solution for the problem. Please check your program and try again.

Language: Java ▾

Type your program below:

```
1 import java.util.*;
2 public class solve
3 {
4     public static void main(String args[])
5     {
6         Scanner scanner = new Scanner(System.in);
7         int a = scanner.nextInt();
8         int b = scanner.nextInt();
9         System.out.println(a-b);
10    }
11 }
```

Compile and Test Submit You are allowed to use any of the following languages: C, C++, Java, Python

2. The logic of the code written here generates the correct output. Thus, the output generated by both “Compile and Test”, and “Submit” are as follows –

Compile and Test, Submit:

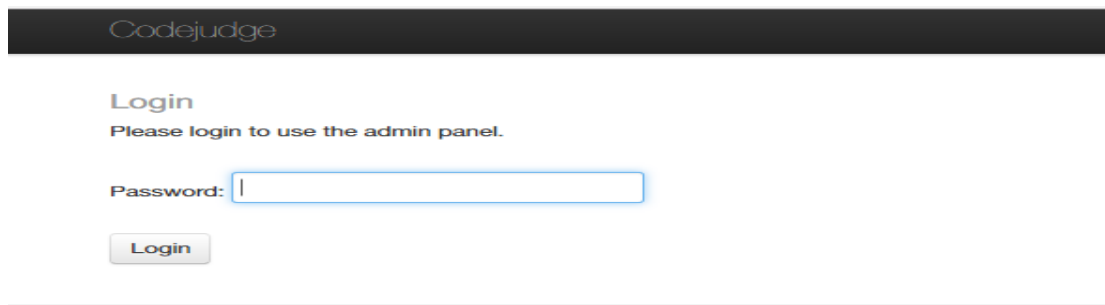
Type your program below:

```
1 import java.util.*;
2 public class solve
3 {
4     public static void main(String args[])
5     {
6         Scanner scanner = new Scanner(System.in);
7         int a = scanner.nextInt();
8         int b = scanner.nextInt();
9         System.out.println(a-b);
10    }
11 }
```

Compile and Test Submit You are allowed to use any of the following languages: C, C++, Java, Python

Congratulations! The sample testcases have passed.

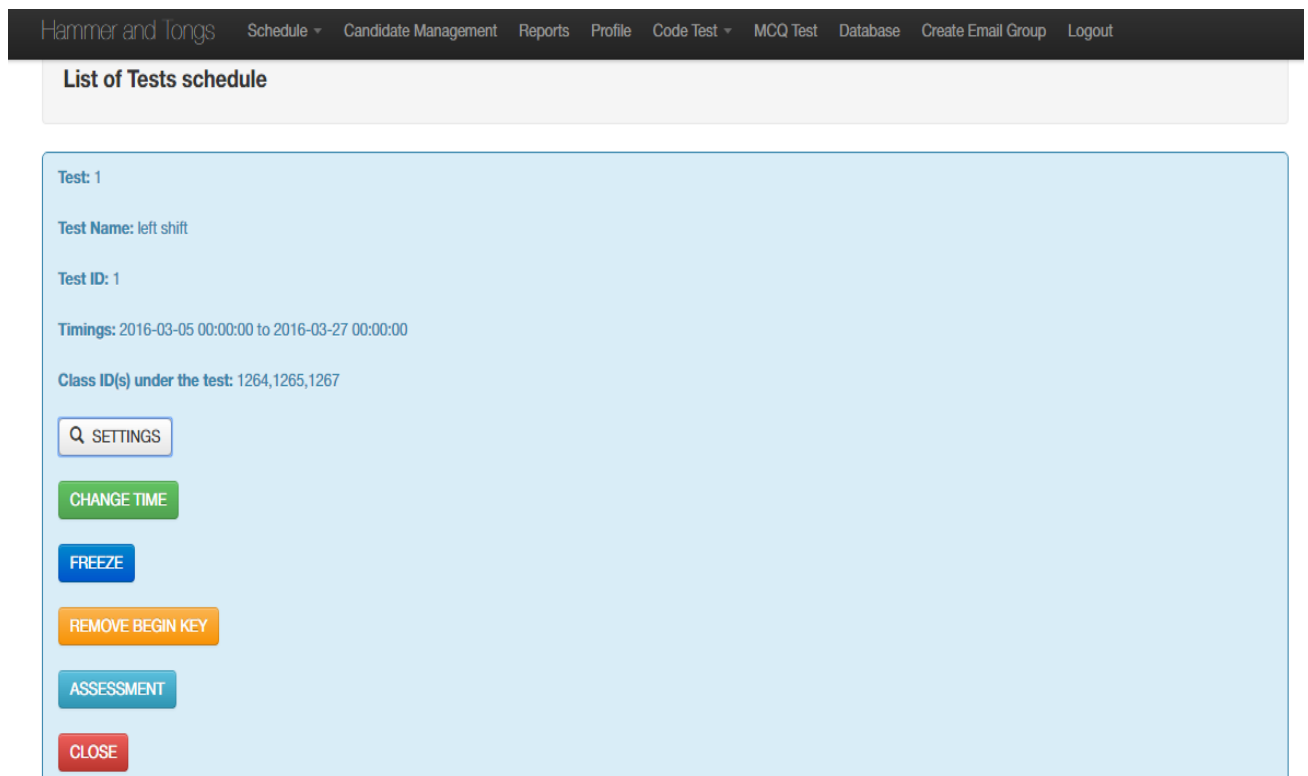
4.2. Admin Login



The image shows the Codejudge Admin Login interface. At the top, there is a dark header with the 'Codejudge' logo. Below the header, the word 'Login' is displayed in a bold, dark font. Underneath, a message reads 'Please login to use the admin panel.' followed by a 'Password:' label and a text input field. A 'Login' button is positioned below the input field.

Since the system has been designed with only one admin, thus, only the password is required to login to the admin account. Now, once the admin log in, there are various actions that he can take.

1. Schedule



The image displays the 'Hammer and Tongs' Admin interface for the 'Schedule' section. The top navigation bar includes links for 'Schedule', 'Candidate Management', 'Reports', 'Profile', 'Code Test', 'MCQ Test', 'Database', 'Create Email Group', and 'Logout'. The main heading is 'List of Tests schedule'. Below this, a light blue box contains details for 'Test: 1':

- Test Name: left shift
- Test ID: 1
- Timings: 2016-03-05 00:00:00 to 2016-03-27 00:00:00
- Class ID(s) under the test: 1264,1265,1267

At the bottom of this box, there is a vertical stack of buttons: 'SETTINGS' (with a magnifying glass icon), 'CHANGE TIME' (green), 'FREEZE' (blue), 'REMOVE BEGIN KEY' (orange), 'ASSESSMENT' (teal), and 'CLOSE' (red).

The schedule option allows the admin to view all the tests that have been scheduled. The snapshot shows the details of one such test. The admin has the privilege to modify the different details related to the test, such as –

- a. Change time: Change the start or end time of the test. Here is a snapshot –

Candidate Management
Reports
Profile
Code Test
MCQ Test
Database
Create

Change Test Timings

Start Time: 2016/08/08 08:00

End Time: 2016/08/08 10:00

CHANGE TEST SCHEDULE

b. Freeze: Using the freeze option, the admin can stop the particular test anytime.

c. Remove Begin Key:

d. Assessment: The admin can view the submissions of all the students who attempted one or more problems in the test, and can also find out whether the submissions have been evaluated by the respective professor, or not. Here is a snapshot -

Students Registered under Test ID :43

Rank	RegNo	Name	Classid	EvalStatus	Marks
1	13BCE1085	Niranj	1264	Not Evaluated	0

2. Candidate Management

The candidate management option allows the admin to view some specific details related to a student, or a professor. The admin can enter the student's email, or the student's registration number, to view the details.

Candidate Management
Reports
Profile
Code Test
MCQ Test
Database
Create Email Group

Manage Candidate

Student Email:

Registration Number:

SEARCH CHANGE PASSWORD

Once the admin enters a particular registration number, or email, the details are displayed as shown in the snapshot given below. Under student test details, the admin can view the tests that have been scheduled for the student, and also, whether the test is still open, or closed. If the test is open, then the number of minutes remaining for the completion of the test is also shown.

Student Details

Registration Number:	13BCE1085
Name:	Niranj
Email Id:	niranj.jyothish2013@vit.ac.in
Recovery Email_Id:	niranj.jyothish2013@vit.ac.in
Block Status:	Unblocked
Classes:	1264 1267

Student Test Details

ID	TEST NAME	STATUS	MINUTES REMAINING
3	Control Flow	CLOSED	0
4	Bit Manipulation	CLOSED	0
43	Stacks and Queues	CLOSED	0
58	Arrays	CLOSED	0
63	Strings	CLOSED	0

3. Create a code Test

The code test option allows the admin to schedule tests for the students. A test must contain one or more problems, so problems also need to be created. Thus, the “create a problem” option is also provided.

a. Create a test

For creating a test, the batch number needs to be entered first. When the “proceed” button is clicked, the class numbers associated with the particular batch are displayed. The admin can choose the classes for which the test needs to be scheduled. The other details that need to be filled in are the test name, start time, end time, and the begin key (if required). Once the test is scheduled, a unique test id is generated.

Batch:

Select class numbers:

☐ 1264 ☒ 1265 ☒ 1266
☐ 1267 ☐ 1268

Title:

Start Time:

End Time:

Key Entered To Begin Test: ☐

b. Creating a problem

A test must contain one or more problems. Now, there are several components associated with a particular problem:-

- i) Test ID - Unique number to indicate to which test the particular problem belongs to.
- ii) Maximum Marks – Maximum marks allocated to the problem.
- iii) Problem title – A title describing the problem.
- iv) Time limit – The execution time limit for the solution program.
- v) Language – Using this option, the admin can restrict the programming language in which the student can submit the solution. If no language is specified, the student is free to choose any of the four languages.

vi) Detailed Problem – This contains a brief description of the problem, and includes the problem statement, the sample input, the sample output etc.

vii) Sample Input, Sample Output – The sample input and sample output, as provided in the detailed problem section, are included here, in the respective sections.

viii) Hidden Input, Hidden Output – The solution submitted by a student need to be evaluated against a set of hidden test cases. Thus, the hidden inputs, and the respective outputs are included in these sections.

viii) Pre-Code, Post-Code (Optional) –Allows the problem setter to add some pre-code or post-code, which cannot be edited by the student.

The screenshot shows a web interface for adding a new problem. At the top, there is a blue header bar with the text "Add problem". Below this, the main form is titled "Add a Problem". The form has several tabs: "Problem", "Sample Input", "Sample Output", "Hidden Input", "Hidden Output", "Pre-Code", and "Post-Code". The "Problem" tab is currently selected. The form contains the following fields and controls:

- Test ID:** A text input field.
- Maximum Marks:** A text input field.
- Problem Title:** A large text input field.
- Time Limit:** A text input field followed by a unit selector set to "ms".
- Language:** A dropdown menu with "Select" as the current value.
- Difficulty Level:** A dropdown menu with "Select" as the current value.
- Detailed problem:** A label followed by a blue button that says "Markdown formatting supported".
- Rich Text Editor:** A large text area with a toolbar above it. The toolbar includes buttons for bold (B), italic (I), link, unlink, list, and image, along with a "Change block type" dropdown menu.

At the bottom of the form, there are two buttons: "Add Problem" (in blue) and "Preview" (in grey).

4. Manage Student Groups

This option deals with creating separate groups for students. If some students end up missing a particular test, then the admin can create a separate group for them, indicating that the test needs to be opened again for this particular group. Such student groups can be created for other purposes as well. As shown in the snapshot below, the details to be entered to create a student

group are – The batch, description of the group, and the registration numbers of the students who belong to the group.

Manage Student Groups

Select a Passout batch(*)

Description:

Batch Registration No. (comma seperated):

13BCE1040, 13BCE1085,
13BCE1146

Create

Here is a snapshot of the groups that have been created –

Group ID	Description	Batch	Edit	Delete
60	Absentees	2017	Edit	Delete
59	Random	2013	Edit	Delete
55	random	2013	Edit	Delete
47	55678	2013	Edit	Delete

The admin can edit the members present in the group, and if the group has served its purpose, then the delete option is provided to remove it.

4.3 FACULTY LOGIN

Faculty Login

Faculty Id:

Password:

The faculty login requires every professor to login through their faculty ID, and a password. Once a professor logs in, he/she will be shown the list of tests scheduled for his/her students. This page is similar to the Schedule page in the Admin account. But, here, the teacher has to evaluate the submissions made by students, and so, on clicking the assessment button, the following page is shown –

Submissions under Test ID :43

#	Class ID	# of submissions	Evaluation Done	Not Done
1	1264	2	0	2

Students Registered under Test ID :43

Rank	RegNo	Name	Classid	EvalStatus	Marks
1	13BCE1085	Niranj	1264	<input type="button" value="Evaluate"/>	0

On clicking the evaluate button, the teacher can view all the submissions made by the particular student, to the problems posted under the given test.

Profile details for 13BCE1085

Details of problems attempted:

Problem	Status
Right Shift	<input type="button" value="Attempted"/>
XOR	<input type="button" value="Attempted"/>
Add two numbers	<input type="button" value="Solved"/>
Subtract two numbers	<input type="button" value="Attempted"/>

Now, the teacher can view the code submitted for a particular problem, by clicking on the problem name. So, when the “Subtract two numbers” link is clicked, the teacher can view –

1. The code submitted by the student.
2. Whether the program gave the right output for the sample inputs, or not.
3. Whether the program gave the right output for the hidden inputs, or not. If the student has not pressed the “submit” button, then the message on this page will indicate that the code hasn’t been evaluated against the hidden inputs. A snapshot -

Subtract two numbers

```
import java.util.*;
public class solve
{
    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        System.out.println(a-b);
    }
}
```

List of Hidden Test Cases are as follows:

Input -
2000
1000

Expected -
1000

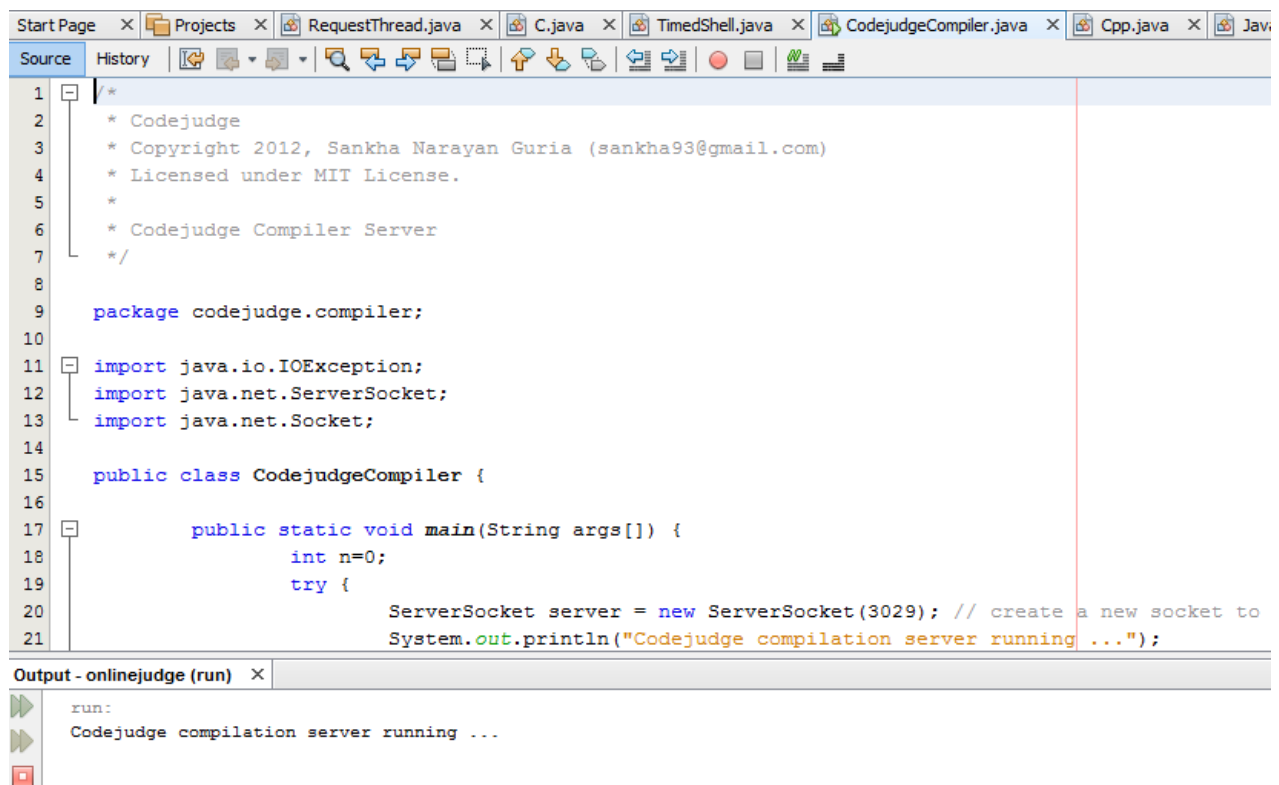
Received -
3000

Failed

The student has passed only sample test cases.

4.4 Java Server

The Java Server has been used to compile and execute the code written, and submitted by the students. When a student presses the “compile and test” or the “submit” button, the code is sent to the java server via a socket. The java server continuously listens to this particular socket, and as soon as a connection is established, it retrieves the code from the buffer, compiles, and executes the code (if there are no compile time errors), and then sends the output back to php through the same socket. PHP, in turn, compares the expected output, to the actual output of the particular program, and displays the relevant message(s) to the student.



The screenshot shows an IDE window with several tabs: Start Page, Projects, RequestThread.java, C.java, TimedShell.java, CodejudgeCompiler.java (active), Cpp.java, and Java. The active tab displays the source code for CodejudgeCompiler.java. The code includes a package declaration, imports for java.io.IOException, java.net.ServerSocket, and java.net.Socket, and a public class CodejudgeCompiler with a main method. The main method initializes a ServerSocket on port 3029 and prints a message. Below the code editor, the 'Output - onlinejudge (run)' window shows the output: 'run: Codejudge compilation server running ...'.

```
1  /*
2   * Codejudge
3   * Copyright 2012, Sankha Narayan Guria (sankha93@gmail.com)
4   * Licensed under MIT License.
5   *
6   * Codejudge Compiler Server
7   */
8
9  package codejudge.compiler;
10
11  import java.io.IOException;
12  import java.net.ServerSocket;
13  import java.net.Socket;
14
15  public class CodejudgeCompiler {
16
17      public static void main(String args[]) {
18          int n=0;
19          try {
20              ServerSocket server = new ServerSocket(3029); // create a new socket to
21              System.out.println("Codejudge compilation server running ...");
```

Output - onlinejudge (run) X

```
run:
Codejudge compilation server running ...
```

CONCLUSION AND FUTURE WORK

This internship proved to be very helpful to me. It proved to be a platform for me, for implementing the new concepts and technologies learnt theoretically and an insight into building enterprise grade software from scratch including the Database schema to the point of deployment and testing on Amazon Web Services.

I was able to get hands on experience on Database designing and implementation and front end technologies like bootstrap and java script and was exposed to many new functionalities of php including mail servers and also got a rich experience of working with parameterized queries. The internship also enabled me to work upon an open source module for compilers and extensively customize it to the organizations needs.

The last part of the internship enabled me to gain hands on experience with AWS (both deployment and testing). Overall, the internship was an enriching experience.

As of now, the portal is hosted on a Windows Server instance running on Amazon Web Services cloud. The system is able to handle a load of about 100 students. In future, the system can be made more scalable using Auto Scaling. Auto Scaling instantiates a new instance of VM when load on existing VMs increases. This will ensure that required numbers of VMs are always running to handle the load.

Load Balancer can be used to distribute the incoming network traffic among the backend servers. If there is excessive traffic, the Load Balancer will divide the traffic among different VMs. This will ensure that VMs will not crash due to inflow of excessive traffic.

Since the AWS IaaS services are paid one, we are trying to host our portal on an in-house cloud. We are trying to install OpenStack on our server to create a cloud environment.

On the application side, apart from code tests MCQ tests options can be added. More work can be done to make the system more secure from external vulnerabilities like DOS attack .

REFERENCES

1. W3Schools: <http://www.w3schools.com/>
2. Stack Overflow: <http://www.stackoverflow.com/>
3. GitHub: <https://github.com/sankha93/codejudge>
4. AWS: <https://aws.amazon.com/>
5. Bootstrap: <http://getbootstrap.com/>
6. PHP: <http://php.net/>
7. Database Design: <https://dbdesigner.net/>
8. Code Editor: <https://codemirror.net/>