**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Title Page** | **Page**  i |
| **Approval Sheet**  **Acknowledgement**  **Abstract** | ii  iii  iv |
| **Table of Contents** | v |
| **List of Tables** | vii |
| **List of Figures** | viii |
|  |  |
| **CHAPTER 1 – INTRODUCTION** |  |
| Background of the Study | 1 |
| Statement of the Problem | 3 |
| Objectives of the Study  Scope and Limitations  Significance of the Study | 3  4  5 |
| Operational Definition of Terms | 5 |
|  |  |
| **CHAPTER 2 – REVIEW OF RELATED LITERATURE AND SYSTEMS** |  |
| Related Literature | 6 |
| Related Systems | 9 |
|  |  |
| **CHAPTER 3 – MATERIALS AND METHODOLOGY** |  |
| Research Design | 14 |
| Project Environment |  |
| Locale | 14 |
| Population of the Study | 14 |
| Research Instruments  Statistical Tools  Timetable | 14  15  15 |
| Data Gathering Procedures | 16 |
| Methodology |  |
| Requirements Specification |  |
| Use Case Diagrams | 17 |
| Use Case Specification | 18 |
| Design |  |
| Project Design | 32 |
| Output and User-Interface Design | 34 |
| Data Design | 40 |
| Development |  |
| Software Specification | 43 |
| Hardware Specification | 43 |
| Program Specification | 43 |
| Deployment Diagram | 43 |
| Test Plan | 44 |
| Test Cases | 45 |
| Verification, Validation and Testing |  |
| Unit Testing | 59 |
| Integration Testing | 59 |
| System Testing  **CHAPTER 4 – RESULTS AND DISCUSSION**  Project Description  Project Structure  Project Capabilities and Limitations  Project Test Results  Test Case  Project Evaluation  Implementation Results  **CHAPTER 5 – CONCLUSIONS AND RECOMMENDATIONS**  Conclusions  Recommendations | 59  60  61  65  66  79  81  81 |
| **References** |  |
| **Appendices** |  |
| 1. Letter of Permission 2. Grammarian’s Certificate |  |
| 1. Statistician’s Certificate |  |
| 1. Survey Questionnaires 2. Questionnaire Validation Sheet |  |
| 1. Research Forum Certificate |  |
| 1. Researchers’ Profile |  |

**List of Tables**

**Table No. Name Page**

1 Level of Interpretation 14

2 Gantt chart (Sep-April 2018) 15

3 Use Case Login 17

4 Use Case Register 18

5 Use Case Create Class 19

6 Use Case Join Class 20

7 Use Case Add Student 21

8 Use CaseDisable Class 22

9 Use CaseRename Class 23

10 Use Case Create Activity 24

11 Use Case Edit Activity 25

12 Use Case Delete Activity 26

13 Use Case Submit activity 27

14 Use Case View Grades 28

15 Use Case Delete Class 29

16 Use Case Remove Student 30

17 Tbl Student 40

18 Tbl Instructor 40

19 Tbl Class 40

20 Tbl Submission 40

21 Tbl Activity 41

22 Tbl ActivityDuration 41

23 Tbl Group 41

24 Tbl Grade 41

25 Software Specification 43

26 Hardware Specification 43

27 Program Specification 43

28 Login 45

29 Register 46

30 Create Class 47

31 Create Activity 48

32 Submit Activity 49

33 View Grade 50

34 Join Class 51

35 Add Student 52

36 Disable Class 53

37 Rename Class 54

38 Edit Activity 55

39 Delete Activity 56

40 Delete Class 57

41 Remove Student 58

42 Test Results (Login) 66

43 Test Results (Register) 66

44 Test Results (Create class) 67

45 Test Results (Create activity) 68

46 Test Results (Submit activity) 69

47 Test Results (View grade) 70

48 Test Results (Join class) 71

49 Test Results (Add student) 72

50 Test Results (Disable class) 73

51 Test Results (Rename class) 74

52 Test Results (Edit activity) 75

53 Test Results (Delete activity) 76

54 Test Results (Delete Class) 77

55 Test Results (Remove student) 78

56 Indicators 79

**List of Figures**

**Figure No. Name Page**

1 Edmodo 9

2 CompileJava 10

3 Skoool 11

4 Codefights 12

5 ProProfs 13

6 Use case model 17

7 Contex Diagram 32

8 Data Flow Diagram (Level 0) 33

9 Register 34

10 Login 34

11 Home 35

12 Activity 35

13 Create Activity 36

14 Grade 36

15 View Grade 37

16 Test Code 37

17 Class 38

18 Create class 38

19 Site Map 39

20 Entity Relationship Diagram 42

21 Deployment Diagram 44

22 Register 61

23 Login 61

24 Home 62

25 Activity 62

26 Create Activity 63

27 Test Code 63

28 Class 64

29 Create class 64

### CHAPTER 1

### INTRODUCTION

### Background of the Study

Technology has revolutionized the way we work and is now set to transform education. Children cannot be effective in tomorrow's world if they are trained in yesterday's skills. Nor should teachers be denied the tools that other professionals take for granted (Blair, 1998). A click of a mouse button provides any student anywhere with unprecedented opportunities to learn. So, if a child in Grand Junction wants to master Japanese, it is possible online. If a budding artist in Five Points wants to study the masterpieces of the Louvre, it's possible online. If a future Stephen Hawking in La Junta wants to study Gravitational Entropy with the man himself, it is possible online. If military parents want continuity in their children's education throughout frequent moves to serve our country, then it is possible online (Paige,2002). With every special newspaper supplement, it seems, those in the business offer new visions, new services we didn't know we needed, yet more exciting equipment and software possibilities that lie just over the horizon and, less well-publicized, an increasing number of routes to what may be educational dead ends (Walker,2001).

Apparently, the development of information Technology and knowledge information society transfer brought huge change of education filed in the world. In fact, Educational infra of Philippines was meager 4-5 years before. Also computing and Internet infra it will not support to education environment. But, Philippine education environment changed fast during 2000-2006 years. Development of network environment is big effect of computer game by Philippine young generation. Also, effect of education reform policy that Philippines government. Philippines is making e-Learning systems for remote education environment. Moreover, there is progressing various project with more interest about e-Learning. Therefore, the study wants to correct analysis and right direction presentation for Philippine remote education environment (Kim, 2007).

In addition, the start of the internet age has changed the scene of our educational system. The Information Age offers a great potential as a learning tool for the people. Nowadays, educational institutions are trying to innovate and are open for the possibility of online learning to deliver cost effective, easily accessible and much interactive learning for all students.

In the 1940s, the first recognizably modern electrically powered computers were created. The limited speed and memory capacity forced programmers to write hand tuned [assembly language](https://en.wikipedia.org/wiki/Assembly_language) programs. It was eventually realized that programming in assembly language required a great deal of intellectual effort and was error-prone. Through the years programmers have developed a more reliable programming language.

As such Java, an object-oriented language similar to C++, was simplified to eliminate language features that cause common programming errors. It is a general-purpose computer programming language that is concurrent and class-based. Today Java is a commonly used foundation for developing and delivering content on the Web. According to Oracle, there are more than 9 million Java developers worldwide and more than 3 billion mobile phones run Java (Java Today, 2015).

In AMA Computer College, researcher found that Java instructors have difficulties to conduct activities, discuss lessons and then check it one by one. Some students tend to not finish the task because of a very short time. Instructors also find recording of grades on their laboratory time only. It is for this reason the researcher found it helpful to have an online way of posting activities, checking of their student’s codes and grading their works.

**Statement of the Problems**

This section introduces the problems observed by the researcher.

1. Teachers for basic Java programming lacks central repository for students’ activities and lesson materials.
2. Teachers of basic Java programming takes time checking students’ laboratory exercises.
3. Students’ grade calculation and recording takes time to be done manually.
4. Access to Java compiler is limited during laboratory time only.

**Objectives of the study**

This study aims to create an online tool for basic Java programming for easy access of students and teachers. Specifically, this study aims:

1. To create an online repository for students’ activities and lesson materials.
2. To create a system that will allow the teacher to check laboratory exercises faster.
3. To create a system that will facilitate grading and recording of laboratory exercises online.
4. To create a system that will provide a Java compiler for students for easy access.

**Scope and Limitations**

This project will simplify the work needed to give activity, record grades and conduct examinations on Java programming subject. This is an online system that helps the students bring laboratory activities on the comfort of their homes, to be submitted on a given date, and leaving the teachers to check it on their free time. The features to be implemented in the system are online repository, file attachment on given activities, submit and resubmit an activity, record graded activities, view progress, create class, export grades, built-in compiler.

**Limitations**

In spite of the items mentioned above there were some unavoidable limitations. First, because activities will be done online it will not be easy for the teachers to control the student’s time of submission and those students who prefer not to submit. Second, other students may tend to forget the activities until the deadline was overdue.

**Significance of the Study**

**User/Client**

The researcher believes that this study will have a contribution on the fields of Java programming and teaching through connecting them online.

**Researcher**

This study will help the researcher improve their skills and understanding in system development.

**Future Researchers**

To help the future researcher have an idea of the current system and use this study as an insight for a new level of improvement.

**Operational Definition of Term**

**Compiler** -A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses.

**Built-in** -Incorporated inside the system.

**JaBa** - stands for “Java basic”.

**Java** -a general-purpose computer programming language designed to produce programs that will run on any computer system.

**Online tool** - is something that uses an Internet connection to access the information it needs.

**Repository** -An online storage of the activities made by the teachers for later use.

**CHAPTER 2**

**RELATED LITERATURE AND SYSTEMS**

**Related Literature**

**World Wide Web**

As the number of users on the World Wide Web increases every day, its use in different areas is also growing. One of the most powerful aspects of the Web is that anybody who has Internet access can browse on the net. This enables sharing the information worldwide (Davidson, 2015).

Likewise, one of the fast-growing areas of the Web is distance education (or distance learning). The reason distance education on the Web is getting popular is because it has advantages over other types of distance education programs. It gives much more flexibility to the users. The users can take the courses they registered for at any computer connected to the Internet. They usually have a more flexible time frame to take their classes and their tests (Kuan, 2015).

Moreover, n terms of programming of these sites developers had until recently to use limited range of technology to choose from. These technologies usually involved Common Gateway Interface (CGI) programming, Javascript, and Microsoft’s Active Server Pages. This paper demonstrates that Java servlets and JDBC can be used programming for these sites (Goodwill, 1999).

**Educational Technology**

Educational technology is the effective use of technological tools in learning. As a concept, it concerns an array of tools, such as media, machines and networking hardware, as well as considering underlying theoretical perspectives for their effective application. It is an inclusive term for the tools and the theoretical foundations for supporting learning and teaching. Educational technology is not restricted to high technology (Sunstar, 2017).

**Learning Management Systems**

Learning management systems range from systems for managing training and educational records to software for distributing [online](https://en.wikipedia.org/wiki/Internet) or blended/hybrid college courses over the Internet with features for online collaboration. Colleges, universities, school districts, and schools use LMSs to deliver online courses and augment on-campus courses. LMSs also act to augment the lessons the teacher is giving in a brick and mortar environment, not just replace them. Corporate training departments use LMSs to deliver online training, as well as to automate record-keeping and employee registration (Manichander, 2016).

In other words, the inappropriate use of LMS in the literature is perhaps most commonly associated with computer applications which are [virtual learning environments](https://en.wikipedia.org/wiki/Virtual_learning_environment) (Course Management Systems, CMS). These systems are used primarily for online or blended learning, supporting the placement of course materials online, associating students with courses, tracking student performance, storing student submissions, and mediating communication between the students as well as their instructor. Some of this same functionality can be seen within LMSs as well, so it is understandable why confusion might exist about the differences between the two types of systems. However, the systemic nature of an LMS does not limit its functionality to that of a CMS. CMS stands for Content Management System. It is a term and has not been endorsed with a solid definition. A CMS can have multiple meanings depending on the scenarios and the person's or project objectives (The Western Design Center, 2018).

**Assessment**

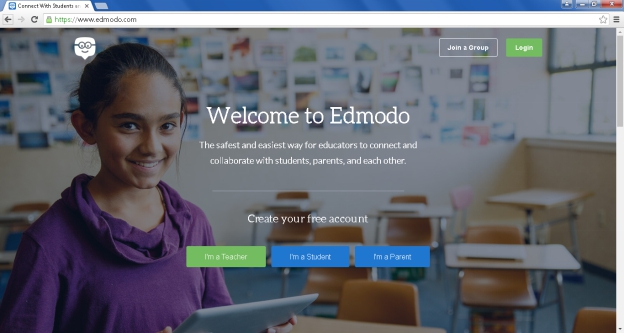
It is always better to have a class that meets any time you are ready and have a computer available, for exactly the amount of time you have, and builds a close relationship with your professor and classmates through collaborative discussions (Gray, 1998; Vetter, 1997). It could be the only way in which students who live far from a campus to take the classes or programs they need.

Moreover, two studies explore how assessment is accomplished in a distance learning environment. One study by the National Center for Education Statistics (Lewis et al., 1997) found that 98% of all institutions use testing for their credit-bearing distance learning classes. One third of the classes used proctored group exams at remote sites, while another third gave proctored exams on campus. About 15% of the instructors sent the students their exams by mail or fax (so the students could take the test independently). About 8% take interactive tests at remote sites using either a computer, video, or telephone. Some of the ramifications of this fast-growing phenomena ([Uzodinma](https://www.researchgate.net/profile/Uzodinma_O), 1999).

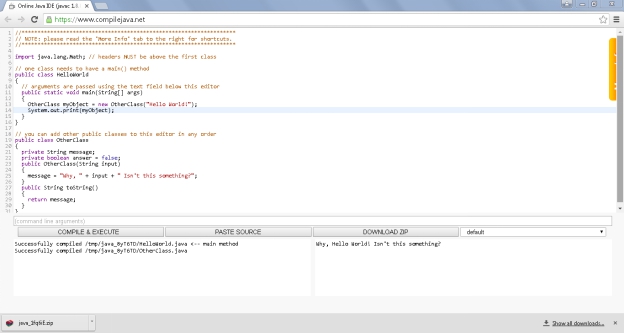
Similarly, the Western Governors University and The University of Phoenix Online use a series of comprehensive examinations to know what classes their students are required to take. Students pass classes by passing only the exams, in some cases. Some of these exams include essay or multiple-choice items, while others are projects. These high stakes, standardized tests are usually administered in proctored, online environments (Carnevale, 2001).

Metropolitan Academy of Manila- School uses the manual way of computing grades and adding information or records of both students and professors. Developing a computerized grading system for Metropolitan Academy of Manila would make the task of recording and computing grades easier for the teachers. This will not only benefit the teachers of the school but will also benefit the students because of the improvement in the accuracy of calculations and in the proficiency and productivity of the teachers. The computers provide convenience to its user through the easier and faster way of recording information. Population of students in almost all-educational institution is rapidly increasing so great demand in teaching force is also becoming higher (Computerized Grading System For Metropolitan Academy of Manila, 2014).

**Related Systems**

Figure 1 **Edmodo**

This study is related to Edmodo.com. It was created by Nic Borg and Jeff O'Hara in 2008.Edmodo is an educational website that takes the ideas of a social network and refines them and makes it appropriate for a classroom. Using Edmodo, students and teachers can reach out to one another and connect by sharing ideas, problems, and helpful tips. A teacher can assign and grade work on Edmodo; students can get help from the entire class on Edmodo. It is a safe environment. There is no bullying or inappropriate content, because the teacher can see everything that is posted on Edmodo. Also parents can join the class to bring a level of transparency that is difficult to achieve without technology. All in all Edmodo is a great companion to just about any class. Just like Edmodo, JABA is an educational website that takes the concept of a classroom into an online web service for the students and teachers.

Figure 2 **CompileJava**

CompileJava was created by WHOISGUARD,inc on October 2013. This present study is related to CompileJava.net (Figure 2) a public website. This website provides an online Java IDE for teachers, students and programmers use. It allows students and teachers to run java programs online without the need of a local java platform. Site is hosted in Ashburn, VA, 20149, United States.



Figure 3 **SKOOOL**

The Skoool site was developed by AIB, The Irish Times and Intel Ireland on 2003. The study is related to Skoool, a public e-learning site. Skoool and the proposed system has similar functions that provides online references for teachers and students to easily access and use it to study. It has an upload/download function in which the person can use as a reference to study a subject.

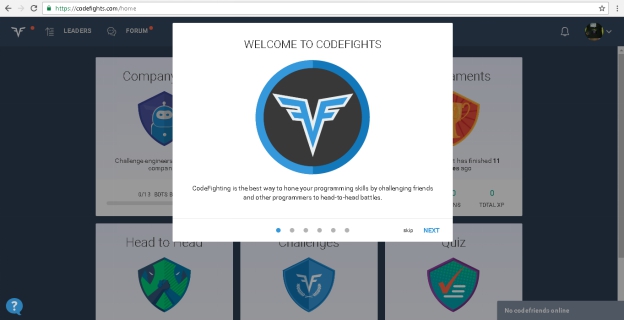


Figure 4 **Codefights**

Figure 4 is codefights, a website for one-on-one competitive programming.

Codefights was developed by Aram Shatakhtsyan and [Tigran Sloyan](https://www.linkedin.com/in/tigransloyan) on 2014 at San Francisco, California, USA.

The study is related to Codefights on developing interest in programming by providing an online compiler.

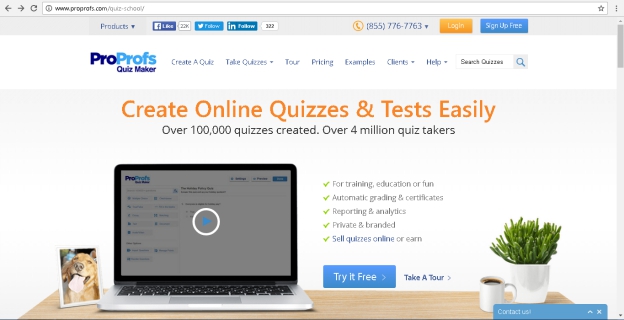


Figure 5 **ProProfs**

ProProfs.com as shown in Figure 5 is a modern, web-based LMS. Founded by Sameer Bhatia on 2005 at Los Angeles, California, USA. Host courses withProProfsor customize the design and embed on your website. This site manages learners through their powerful Learning Management System (LMS) software. Overview ofProProfsLMS Features, creates courses & e-learning content.

The study is similar to the proposed system since both systems let you create and deliver online activities. The proposed system and Proprofs both have an automated grading on online activities.

**Chapter 3**

**MATERIALS AND METHODOLOGY**

This chapter discusses the materials and methodology used by the researcher in gathering data for the proposed study in AMA Computer College Davao.

**Research Design**

The proponent utilized the Descriptive Research Design. Descriptive Research Design mainly is a scientific method which involves observing and describing the behavior of a subject without influencing it in any way(Shuttleworth,2008). The proponent applied the interview method in gathering the data that was used in creating the system.

**Project Environment**

**Locale**

The proponent has conducted the research study at AMA Computer College Davao located at 123 General Malvar Street Davao City.

**Population of the Study**

There are forty (40) respondents of this study consisted of one instructor who teaches Java programming and 39 2nd year Java Programming students.

**Research Instruments**

**Survey questionnaire**

The proponent used a validated questionnaire in gathering data. A questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from the respondents. Although they are often designed for statistical analysis of the responses, this is not always the case.

**Interview questionnaire**

An interview guide was also used to ascertain problems encountered both by the students and teachers.

**Table 1 Level of interpretation**

|  |  |  |
| --- | --- | --- |
| **Range** | **Descriptive Level** | **Interpretation** |
| 4.20– 5.00 | Strongly Agree | The specified feature function is highly acceptable to address the need or issue. |
| 3.40 – 4.19 | Agree | The specified feature function is acceptable to address the need or issue. |
| 2.60 – 3.39 | Neither Agree or  Disagree | The specified feature function may or may not address the need or issue |
| 1.80 – 2.59 | Strongly Disagree | The specified feature function is slightly unacceptable to address the need or issue. |
| 1.00 – 1.79 | Disagree | The specified feature function does not really address the need or issue. |

Table 1 shows the level of interpretation that was used in analyzing the data gathered.

**Statistical Tools**

The gathered data was tallied and interpreted using the mean scores.

**Mean** is the average of all numbers and is sometimes called the [arithmetic mean](http://whatis.techtarget.com/definition/arithmetic-mean). To calculate mean, add together all of the numbers in a set and then divide the sum by the total count of numbers.

**Timetable**

**Table 2 Gantt chart (Sep 2017-April 2018)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** | **JABA: Online tool for basic Java programming** | | | | | | | | | | | | | | | | | | | | | | | | |
| SEP | | | OCT | NOV | | | | | | DEC | | | JAN | FEB | | | MAR | | | APR | | | | |
| Research |  | | |  |  | |  | |  | | |  | | | |  | | |  | | |  | |  | |
| Planning |  |  | | | | | |  | |  | | |  | | | |  | | |  | | |  | |  |
| Documentation |  |  |  | | | | | | | | | | | | | | | | | | | | | |  |
| Design |  |  |  | | |  | | | | | | | | | | | | | | | | | | |  |
| Implementation |  |  |  | | |  | |  | | | | | | | | | | | | | | | | |  |
| Deployment |  |  |  | | |  | |  | |  | | |  | | | |  | | |  | | | | |  |

This table shows the progress of the study.

**Research** - In this time, the researcher investigated ideal subjects to study for their capstone/thesis.

**Planning-** After the researcher found the subject, the researcher begun planning the steps and procedures appropriate to be used for the capstone/thesis.

**Documentation** – The researcher documented the procedures and steps the researcher did for the study. The researcher also documented the data gathered and analyzed for the study.

**Implementation** - Once the system is completed and ready to use the researcher will deploy the system in their chosen school.

**Data Gathering Procedures**

1. The researcher, asked permission from the school management of AMA Computer College Davao to conduct the study.
2. Then, he conducted interview with the instructor teaching Java programming.
3. Next, the interview was transcribed for validation.
4. Lastly, the researcher presented the interview transcript to the interviewee for validation

**Methodology**

**Requirements Specification**

**Use Case Diagram**

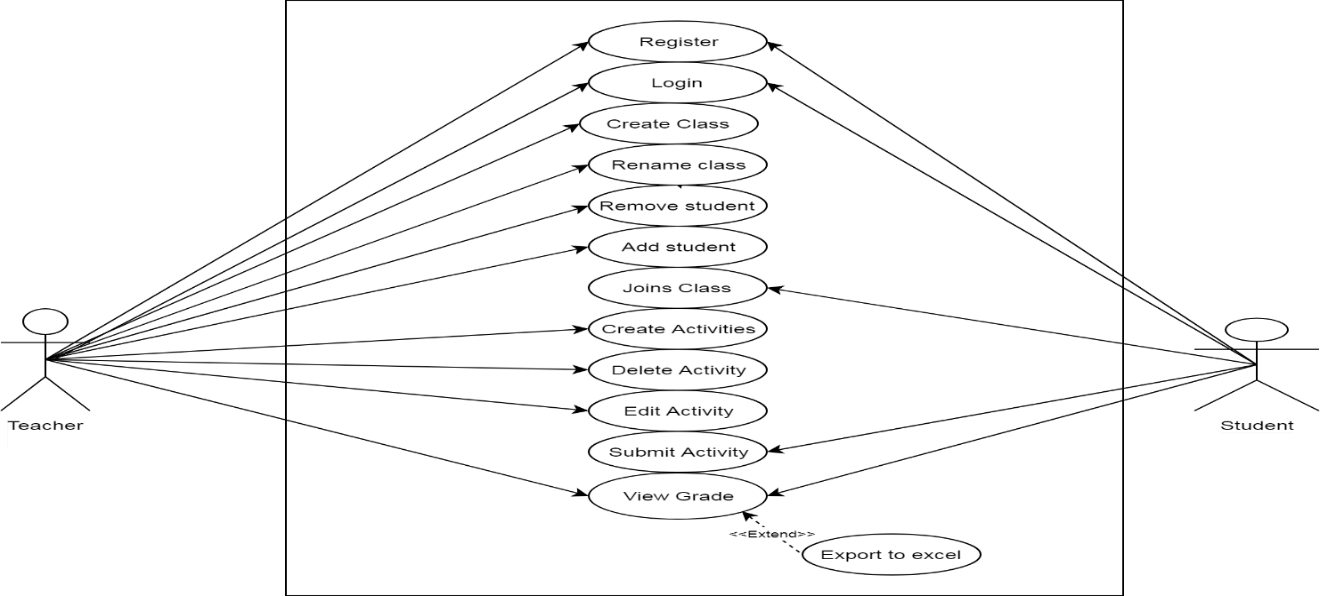


Figure 6 **Use case model**

The use case model in Figure 6 shows the interaction of the users inside the system. The use cases, are the specific roles played by the actors (users) within and around the system.

**Use case specification**

**Table 2 - Use Case Login**

|  |  |
| --- | --- |
| ID | UC-001 |
| Title: | Login |
| Description: | The user will login his username and password |
| Primary Actor: | Teacher, student |
| Preconditions: | 1. There must be internet connection.  2. The user must be registered. |
| Post conditions: | User is directed to home page |
| Main Success Scenario: | 1. The user clicks the login button. |
| 1. The login form appears. |
| 1. The user inputs username and password |
| 1. The user clicks login button   4.1 The system validates the inputs |
| Alternative Flows: | 1. Invalid inputs on required information. |
| 1.1 Displays an error message. |
| Include/Extensions: | None |
| Frequency of Use: | Every time the user needs to access the site. |

**Table 3- Use Case Register**

|  |  |
| --- | --- |
| ID | UC-002 |
| Title: | Register |
| Description: | The user will register personal information |
| Primary Actor: | teacher, student |
| Preconditions: | 1. There must be internet connection. |
| Post conditions: | User may now login |
| Main Success Scenario: | 1. The user clicks the register button. |
| 1. The registration form appears. |
| 1. The user inputs basic information   3.1 The system validates each input |
| 1. The user submits his/her registration form. |
| Alternative Flows: | 1. Invalid inputs on required information. |
| 1.1 Displays an error message. |
| Include/Extensions: | None |
| Frequency of Use: | User registers for an account. |

**Table 4- Use Case Create Class**

|  |  |
| --- | --- |
| ID | UC-003 |
| Title: | Create class |
| Description: | The user will create a class for the students |
| Primary Actor: | Teacher |
| Preconditions: | The user’s account should be a teacher. |
| Post conditions: | 1. A class is created. 2. A code is given to the user for the students to access the class. 3. The students may now join the class. |
| Main Success Scenario: | 1. The user clicks create class button. |
| 1. The user inputs a class name. 2. System validates the class name. |
| 1. The user clicks submit button. |
| Alternative Flows: | 3. Invalid class name. |
| 3.1 Displays an error message. |
| Include/Extensions: | None |
| Frequency of Use: | Every time the teacher need to create a class |

**Table 5- Use Case Join Class**

|  |  |
| --- | --- |
| ID | UC-004 |
| Title: | Join class |
| Description: | The user will join a class. |
| Primary Actor: | Student |
| Preconditions: | 1. The user should have the class code  2. The user must be logged in. |
| Post conditions: | 1. The user successfully joined the class. |
| Main Success Scenario: | 1. The user inputs the code to search for the class.  1.1 System searches for the code’s corresponding class. |
| 1. The user clicks join class button.    1. A dialog box appears (“You are about to join a class.”). |
| 1. User clicks ok |
| Alternative Flows: | 3. User clicks cancel |
| 3.1 cancels the action |
| Include/Extensions: | None |
| Frequency of Use: | Every time a student needs to join a class. |

**Table 6- Use Case Add student**

|  |  |
| --- | --- |
| ID | UC-005 |
| Title: | Add student |
| Description: | The teacher adds some students on the class created. |
| Primary Actor: | Teacher |
| Preconditions: | The student should be a registered member. |
| Post conditions: | 1. Students are successfully added 2. The user must be logged in. |
| Main Success Scenario: | 1. The user selects students to be a member of the class. |
| 1. The system displays every student registered to the teacher’s previous classes. |
| 1. The user clicks add button. |
| 1. sMessage box displays. (“Student added successfully”) |
| Alternative Flows: | 1. Student already a member |
| 1.1 Displays a message. |
| Include/Extensions: |  |
| Frequency of Use: | When the teacher creates a class. |

**Table 7- Use Case Disable class**

|  |  |
| --- | --- |
| ID | UC-006 |
| Title: | Disable class |
| Description: | Disable a class that is not in use. |
| Primary Actor: | Teacher |
| Preconditions: | 1. The teacher should be the one who created the class to be disabled 2. The user must be logged in. |
| Post conditions: | Class is disabled. |
| Main Success Scenario: | 1. The user clicks class on the navigation bar.   1.1 System displays all classes created by the user. |
| 1. The user selects disable button on the option menu.    1. The system disables the class. |
| 1. System displays a success message. |
| Alternative Flows: | None |
| Include/Extensions: | none |
| Frequency of Use: | When a class is to disabled. |

**Table 8- Use Case Rename class**

|  |  |
| --- | --- |
| ID | UC-007 |
| Title: | Rename class |
| Description: | The teacher renames a class |
| Primary Actor: | Teacher |
| Preconditions: | 1. The teacher should’ve already created a class. 2. The user must be logged in. |
| Post conditions: | The class name is changed |
| Main Success Scenario: | 1. The user clicks the class name.   1.1 The user clicks rename class button |
| 1. The user inputs a new class name   2.1 The system checks if the name is available  2.2 user clicks ok button. |
| Alternative Flows: | 2.1 Class name not available |
| 2.1.1 System displays an error message. |
| Include/Extensions: |  |
| Frequency of Use: | When the teacher wants to rename a class |

**Table 9- Use Case Create activity**

|  |  |
| --- | --- |
| ID | UC-008 |
| Title: | Create activity |
| Description: | The user creates an activity. |
| Primary Actor: | Teacher |
| Preconditions: | 1. The user must be logged in. |
| Post conditions: | An activity is created. |
| Main Success Scenario: | 1. The user clicks the create activity button.   1.1 The user is directed to the create activity form. |
| 1. The user inputs name, duration and activity details.   2.1 The system validates the inputs. |
| 1. The user clicks submit button. |
| Alternative Flows: | None |
| Include/Extensions: | None |
| Frequency of Use: | Every time the user wants to create an activity. |

**Table 10- Use Case Edit activity**

|  |  |
| --- | --- |
| ID | UC-009 |
| Title: | Edit activity |
| Description: | The teacher edits an activity |
| Primary Actor: | Teacher |
| Preconditions: | 1. The teacher have already created an activity. 2. The user must be logged in. |
| Post conditions: | The activity is altered. |
| Main Success Scenario: | 1. The user selects an activity. |
| 1. The user clicks edit. |
| 1. The user edits previous inputs. |
| 4. The system validates for invalid input. |
| 5. user clicks submit button. |
| Alternative Flows: | None |
| Include/Extensions: | None |
| Frequency of Use: | When the teacher wants to edit an activity |

**Table 11- Use Case Delete activity**

|  |  |
| --- | --- |
| ID | UC-010 |
| Title: | Delete activity |
| Description: | The teacher deletes an activity. |
| Primary Actor: | Teacher |
| Preconditions: | 1. The teacher have already created an activity. 2. The user must be logged in. |
| Post conditions: | The activity is deleted. |
| Main Success Scenario: | 1. The user selects an activity. |
| 1. The user clicks delete. |
| 1. The system displays a message   (“All information about this activity will be lost, do you want to continue?”?”). |
| 1. The user clicks ok button. |
| 1. The system deletes the selected activity. |
| Alternative Flows: | None |
| Include/Extensions: | None |
| Frequency of Use: | When the teacher wants to delete an activity |

**Table 12- Use Case Submit activity**

|  |  |
| --- | --- |
| ID | UC-011 |
| Title: | Submit activity |
| Description: | The user submits an activity. |
| Primary Actor: | Student |
| Preconditions: | 1. The activity should not have expired. 2. The user must be logged in. |
| Post conditions: | Activity submitted. |
| Main Success Scenario: | 1. The user clicks the answer activity button.   1.1 The website redirects to the answer activity page. |
| 1. The user types the answer on the online compiler |
| 1. The system compiles source code if answer matches the expected input and output. |
| 1. User clicks submit button. |
| 1. The system saves the grade (Pass/Fail) of the student. |
| Alternative Flows: | None |
| Include/Extensions: | 1. The user edits his answers on an activity.   1.1 Resubmits the activity. |
| Frequency of Use: | Every time the user answers an activity. |

**Table 13- Use Case View grades**

|  |  |
| --- | --- |
| ID | UC-012 |
| Title: | View grades |
| Description: | The user views graded activities. |
| Primary Actor: | Teacher, student |
| Preconditions: | There should be graded activities in a class. |
| Post conditions: | Export grades |
| Main Success Scenario: | 1. The user clicks grades on the navigation bar. |
| 1. The system redirects to the grade page. |
| 1. The user selects a class. |
| 1. The system will display the following:    1. (Student) a table is displayed with al the activity grades.    2. (Teacher) a table is displayed with all the students’ grades. |
| Alternative Flows: | None |
| Include/Extensions: | * + 1. User clicks export grade button.     2. System donwloads the grade table on local computer. |
| Frequency of Use: | Every time the user wants to view grades. |

**Table 14- Use Case Delete Class**

|  |  |
| --- | --- |
| ID | UC-013 |
| Title: | Delete class |
| Description: | The user deletes the class’s excising files. |
| Primary Actor: | Teacher |
| Preconditions: | 1. user must be logged in. |
| Post conditions: | The class’s files are deleted. |
| Main Success Scenario: | 1. The user clicks class on the navigation bar. |
| 1. System displays all classes created by the user. |
| 1. User selects a class. |
| 1. The user selects delete files button on the option menu.    1. The system displays a message (“Are you sure you want to delete the files on this class?”). |
| 1. The user clicks ok |
| 1. The system deletes all the records in the class. |
| Alternative Flows: | None |
| Include/Extensions: | None |
| Frequency of Use: | Whenever the user wants to clear the records of a class. |

**Table 15- Use Case Remove student**

|  |  |
| --- | --- |
| ID | UC-014 |
| Title: | Remove student |
| Description: | The user removes a student on the class |
| Primary Actor: | Teacher |
| Preconditions: | 1. user must be logged in. |
| Post conditions: | The student have been removed. |
| Main Success Scenario: | 1. The user clicks class on the navigation bar. |
| 1. System displays all the classes created by the user. |
| 1. The user clicks view students. |
| 1. The system displays all students registered on the class. |
| 1. The user selects a student. |
| 1. The user clicks remove student. |
| 1. The system will display a message “Are you sure?”. |
| 1. User clicks ok. |
| 1. The system removes the student from the class table. |
| Alternative Flows: | None |
| Include/Extensions: | None |
| Frequency of Use: | Whenever the user wants to clear the records of a class. |

**Design**

This section contains the following designs of the system: Project Design, Output and User-interface Design, and Data Design.

**Project Design**

This section includes the Context Level Data Flow Diagram, and the Data Flow Diagram.



Figure 7-**Context Level Data Flow Diagram**

This figure shows the information of the system and the processes participating of the external entities of the user.

**Data Flow Diagram**

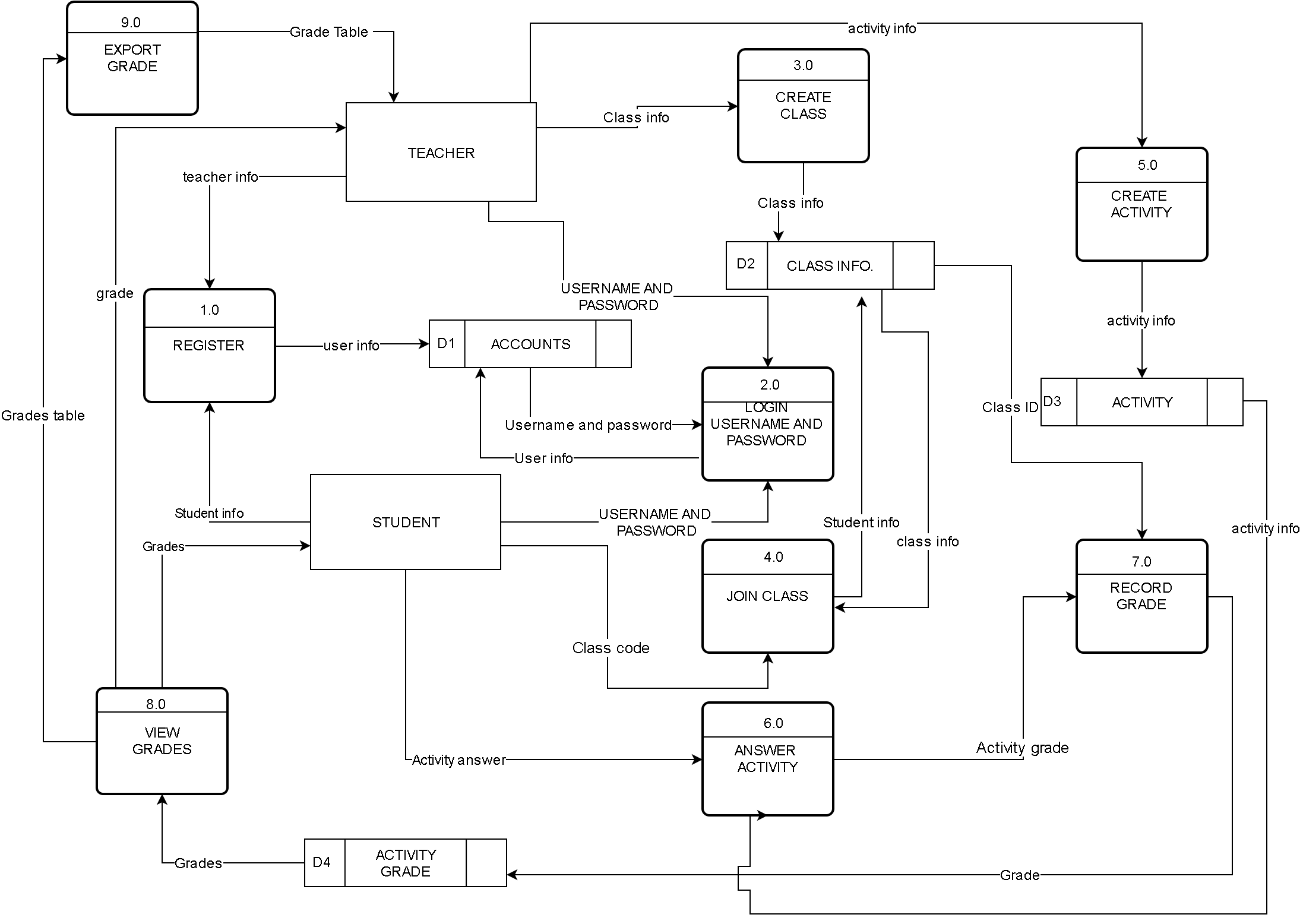
****

Figure 8-**Data Flow Diagram (Level 0)**

Figure 8 shows graphical presentation of the flow of data that the user must execute. It shows the overview of the system and the details that are used in each process.

**User Interface**

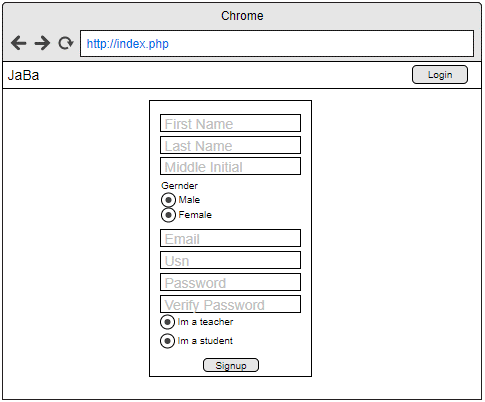


Figure 9- **Register**

Figure 9 presents the registration form of the website.

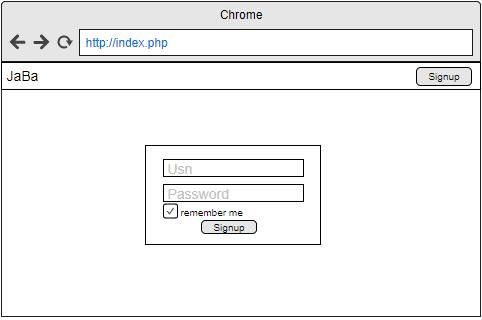


Figure 10- **Login**

Figure 10 shows the login form needed to access the website.

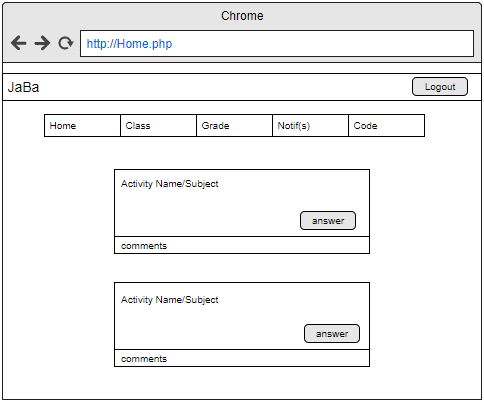


Figure 11- **Home(Student)**

Figure 11 exhibits the home page after student logging in.

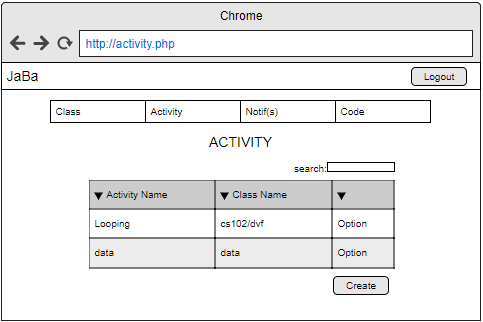


Figure 12**- Activity(Teacher)**

Figure 12 shows all the activities made by the teacher.

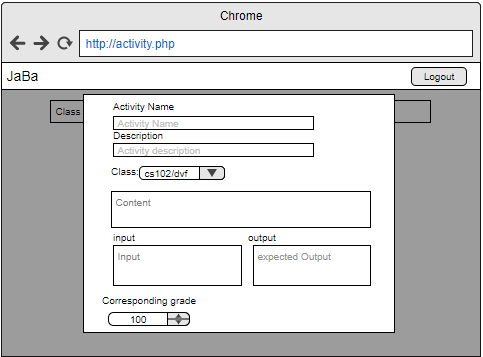


Figure 13- **Create activity(Teacher)**

Figure 13 displays the create activity form.

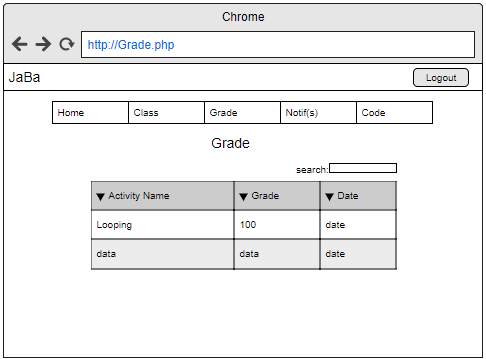


Figure 14**- Grade(Student)**

Figure 14 shows the list of grades of the student by class.

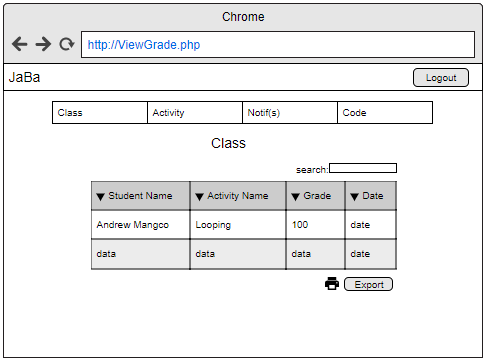


Figure 15- **View grade (Teacher)**

Figure 15 exhibits the grades of the students on a certain activity.

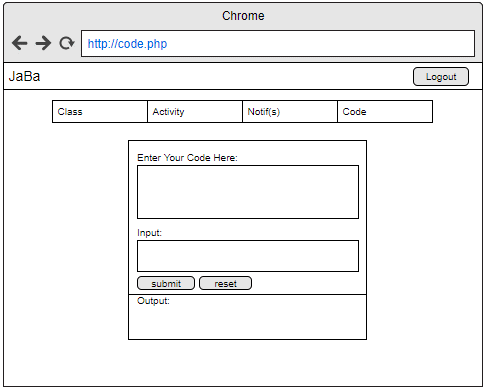


Figure 16**- Test Code**

Figure 16 shows the code testing page.

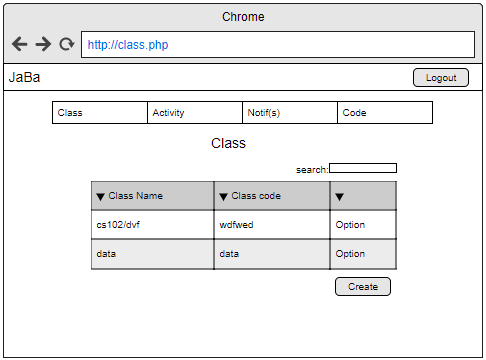


Figure 17**-Class(Teacher)**

Figure 17shows class viewing.

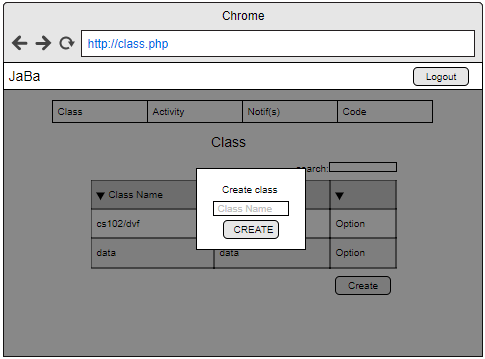
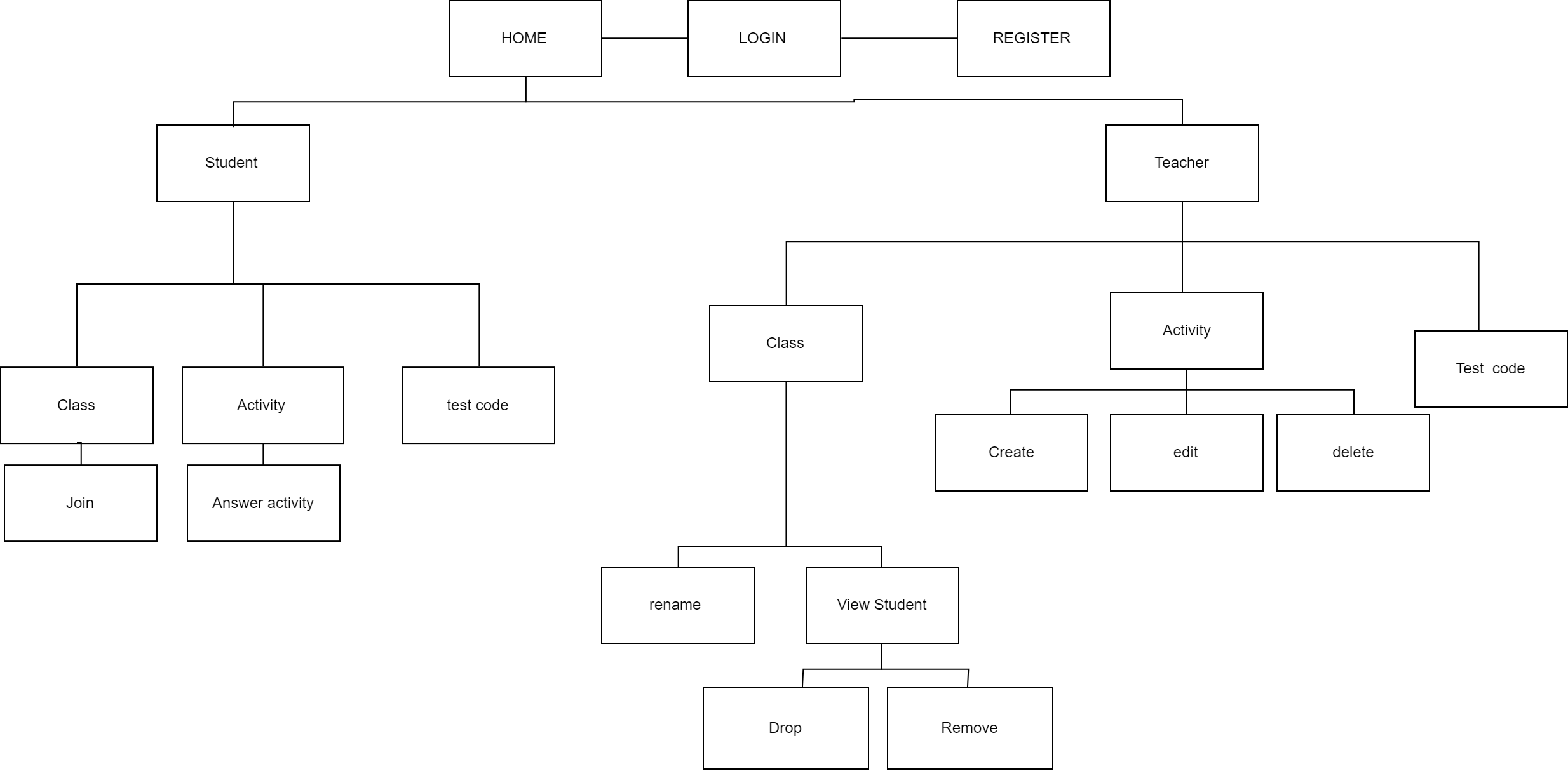


Figure 18**- Create Class**

Figure 18 displays how a teacher could create a class.

**Menu Hierarchy**

This figure shows the overall menu or the system.

****

**HHdwaawdaawdawdadawdaddadawdad**Figure 19**- Site Map**

Figure 19 displays the overall process of the system. The map starts from registration. The user may access the website through logging in as a teacher or as a student. The user may access other pages and options of the website.

**Data dictionary**

Table 20- TBL STUDENT

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | studentId | varchar(11) | no |
|  | fname | varchar(30) | yes |
|  | lname | varchar(30) | yes |
|  | mi | varchar(2) | yes |
|  | gender | varchar(6) | yes |
|  | password | varchar(30) | yes |
|  | email | varchar(30) | yes |

Table 21- TBL INSTRUCTOR

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | instructorId | varchar(11) | yes |
|  | lname | varchar(30) | yes |
|  | fnam | varchar(3) | yes |
|  | mi | varchar(2) | yes |
|  | gender | varchar(30) | yes |
|  | password | varchar(30) | yes |
|  | email | varchar(50) | yes |

Table 22- CLASS

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | classId | varchar(10) | no |
| FK | instructorId | varchar(11) | no |
|  | className | varchar(30) | no |
|  | classCode | varchar(6) | no |

Table 23- joinClass

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NUM |
| PK | className | varchar(20) | no |
| FK | studentId | varchar(11) | no |

Table 24- activity

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | activityId | varchar(10) | no |
|  | Description | varchar(500) | yes |
|  | createDate | timestamp | no |
| FK | className | varchar(20) | no |
|  | input | varchar(50) | no |
|  | output | varchar(50) | no |
|  | activityName | varchar(30) | no |
|  | Corresponding gade | int(3) | no |
|  | content | varchar(500) | no |
|  | expiration | date | no |
|  | active | varchar(3) | no |

Table 25- activity created

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | activityName | varchar(30) | no |
| FK | instructorId | varchar(11) | no |

Table 26- GRADE

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX KEY | COLUMN NAME | TYPE | NULL |
| PK | activityName | varchar(30) | no |
| FK | studentId | varchar(11) | no |
| FK | grade | int(3) | no |

**Entity Relationship Diagram**

Figure 20 shows data model of the system. The figure illustrates the relationship of the data tables inside the system.

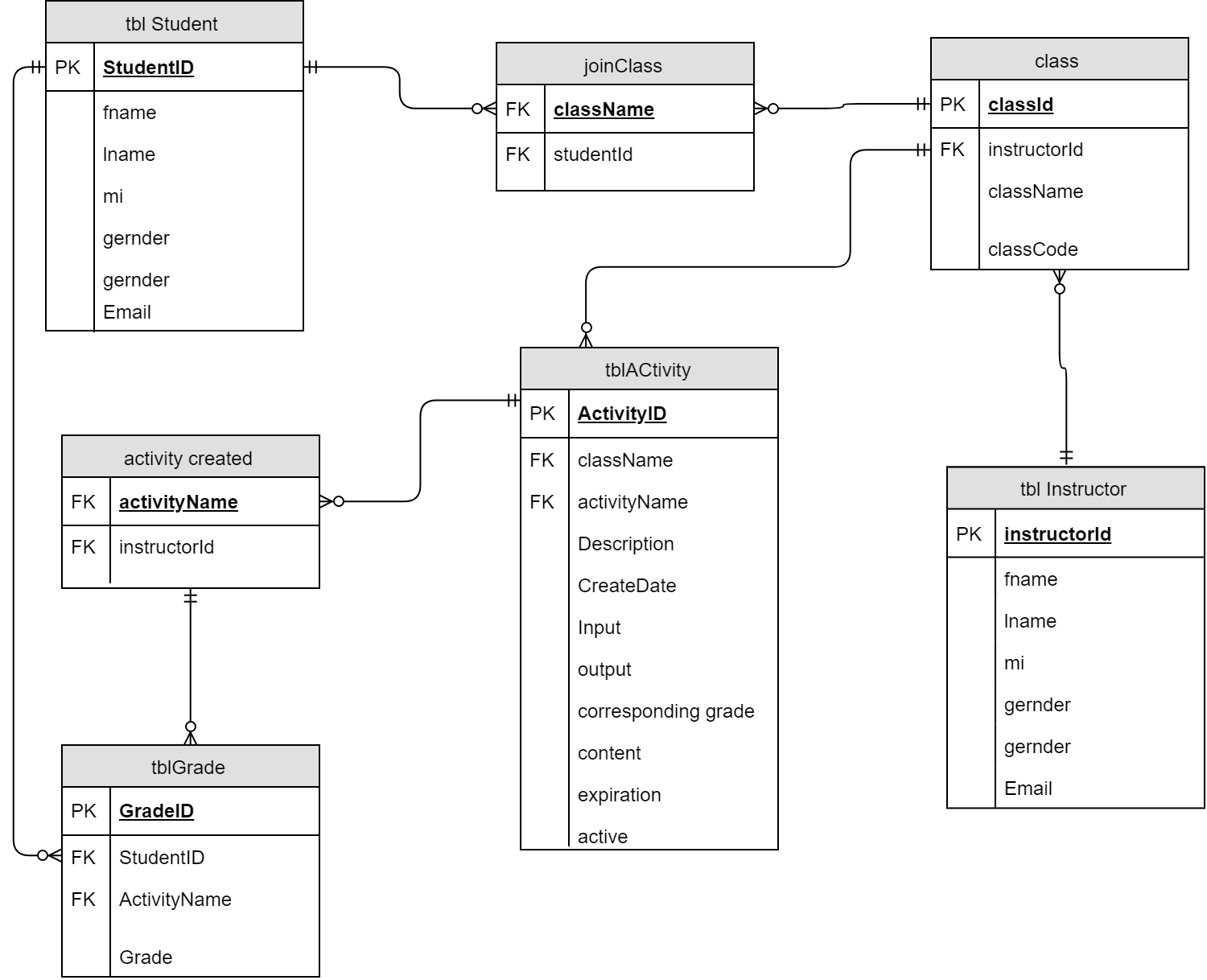
****

Figure 20-**Entity Relationship Diagram**

**Development**

**Software Specification**

The following are the list of software that will be used throughout the implementation.

**Table 28- Software Specification**

|  |  |
| --- | --- |
| 1. Browser |  |
| 1. Java runtime environment | |

This table shows the Software needed to develop the system.

**Hardware Specification**

These are the minimum hardware requirements needed to run the system:

**Table 29- Hardware specification**

|  |
| --- |
| 1. Processor : AMD a4-6300 3.7GHz |
| 1. At least 2 Gigabytes(GB) of RAM |

**Program Specification**

This table displays the programming languages, software and databases that were used by the researcher to meet the systems requirements.

**Table 30- Program specification**

|  |
| --- |
| 1. Programming languages : JavaScript,CSS,PHP |
| 1. Database : Mysql |

**Deployment Diagram**

The deployment diagram specifies a set of constructs that can be used to define the execution architecture of systems that represent the assignment of software artifacts to nodes. Nodes are connected through communication paths to create network systems of arbitrary complexity.

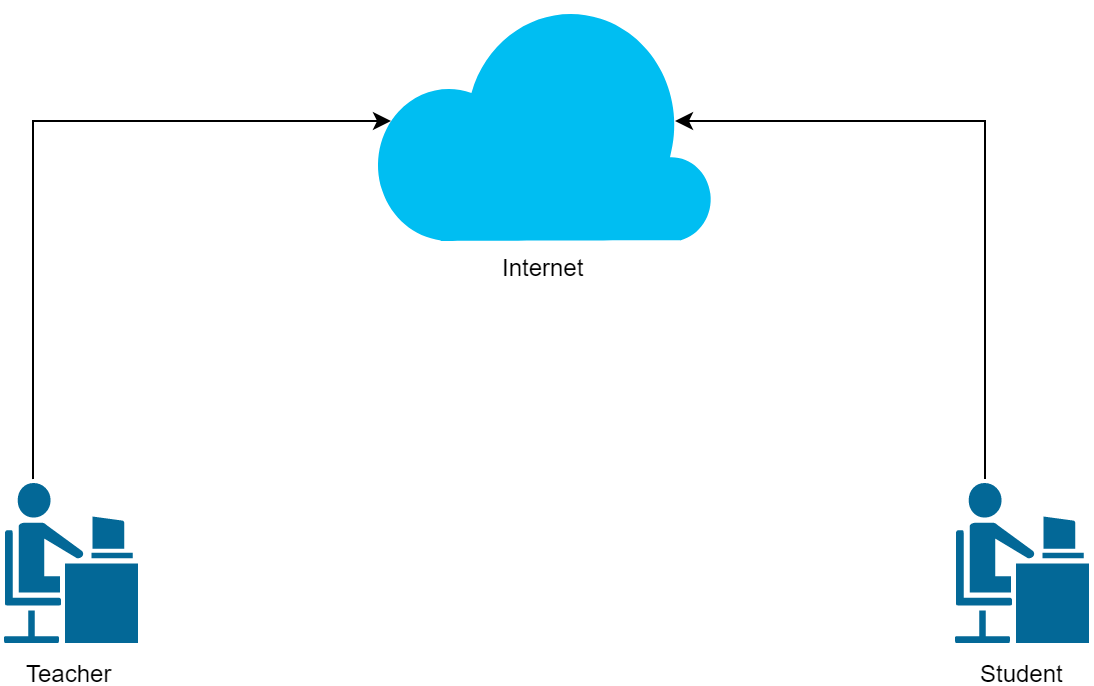


Figure 21-**Deployment Design**

**Test Plan**

**Test Cases**

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

**Table 31- Login**

|  |  |
| --- | --- |
| **Test Case:** | TC-1.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher and Student |
| **Description:** | The user logins his username and password |
| **Test Case name:** | Login |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: The user must be registered.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | User click sign in button | Sign in form appears. |  |  |
| 2 | Input username and Password | System validates username and password. |  |  |
| 3 | User click submit | Website will change its status to logged. |  |  |
| Post-Condition:Website will be available for access. | | | | |

**Table 32- Register**

|  |  |
| --- | --- |
| **Test Case:** | TC-2.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher and Student |
| **Description:** | The user registers his personal information |
| **Test Case name:** | Register |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: There should be internet connection.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | Click register button. | The system will registration form appears. |  |  |
| 2 | Inputs required information | The system checks for invalid inputs. |  |  |
| 3 | Click submit | System displays “Registered successfully!” |  |  |
| Post-Condition: User may now login. | | | | |

**Table 33-Create class**

|  |  |
| --- | --- |
| **Test Case:** | TC-3.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The user create an online class |
| **Test Case name:** | Create class |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: Teacher must be logged in to create class** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | User click sign in button | Sign in form appears. |  |  |
| 2 | Input class name | System validates the class name. |  |  |
| 3 | User click submit | A messagebox appears with the class code. |  |  |
| Post-Condition:   1. A class is created. 2. A code is given to the user for the students to access the class. 3. The students may now join the class. | | | | |

**Table 34- Create Activity**

|  |  |
| --- | --- |
| **Test Case:** | TC-4.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The user creates an online activity |
| **Test Case name:** | Create Activity |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: Teacher must be logged in to create class** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | Click create activity button | System redirects to create activity form. |  |  |
| 2 | The user inputs activity information | System makes sure the input and output of the quiz is filled. |  |  |
| 3 | Click Create | A new activity is created successfully. |  |  |
| Post-Condition:   1. A class is created. 2. A code is given to the user for the students to access the class. 3. The students may now join the class. | | | | |

**Table 35-Submit Activity**

|  |  |
| --- | --- |
| **Test Case:** | TC-5.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Student |
| **Description:** | The user submits an activity |
| **Test Case name:** | Submit Activity |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: Student must be logged in to submit an activity** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | click activity button | The website redirects to the answer activity page. |  |  |
| 2 | The user types the answer on the online compiler | The system compiles source code if answer matches the expected input and output. |  |  |
| 3 | User clicks submit button. | The system saves the grade (Pass/Fail) of the student. |  |  |
| Post-Condition: The activity has been submitted. | | | | |

**Table 36- View grade**

|  |  |
| --- | --- |
| **Test Case:** | TC-6.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Student, Teacher |
| **Description:** | The user views grades |
| **Test Case name:** | View grade |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition: User must be logged in to submit an activity** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks grades on the navigation bar. | The system redirects to the grade page. |  |  |
| 2 | The user selects a class. | 1. The system will display the following:    1. (Student) a table is displayed with al the activity grades.    2. (Teacher) a table is displayed with all the students’ grades. |  |  |
| 3 | User clicks export file button. | The system exports the graded table to local computer. |  |  |

**Table 37- Join Class**

|  |  |
| --- | --- |
| **Test Case:** | TC-7.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Student |
| **Description:** | The user Joins a class. |
| **Test Case name:** | Join Class |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**   * + - 1. **Teacher must have created a class**       2. **The user should have the class code** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user inputs the code to search for the class. | System searches for the code’s corresponding class. |  |  |
| 2 | The user clicks join class button. | A dialog box appears (“You are about to join a class.”). |  |  |
| 3 | User clicks ok | System adds the student to the class. |  |  |
| Post-Condition:The user successfully joined the class. | | | | |

**Table 38- Add student**

|  |  |
| --- | --- |
| **Test Case:** | TC-8.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The user adds a student to a class. |
| **Test Case name:** | Add student |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:The student should be a registered member.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects students to be a member of the class. | The system displays every student registered to the teacher’s previous classes. |  |  |
| 2 | The user clicks add button.  . | System updates the student table. |  |  |
| Post-Condition:Student successfully added to class. | | | | |

**Table 39- Disable Class**

|  |  |
| --- | --- |
| **Test Case:** | TC-9.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | Disable a class that is not in use. |
| **Test Case name:** | Disable Class |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**  **1. User must be logged in as teacher**  **2. The teacher should be the one who created the class to be disabled.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. |  |  |
| 2 | The user selects disable button on the option menu.  . | The system disables the class. |  |  |
| Post-Condition:Class is disabled. | | | | |

**Table 40- Rename Class**

|  |  |
| --- | --- |
| **Test Case:** | TC-10.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The teacher renames a class |
| **Test Case name:** | Rename class |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**  **The teacher should’ve already created a class.**  **The user must be logged in.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks rename class button. | The system checks if the desired name is available |  |  |
| 2 | User clicks submit button. | The system changes the name of the class. |  |  |
| Post-Condition:Class is renamed. | | | | |

**Table 41-Edit activity**

|  |  |
| --- | --- |
| **Test Case:** | TC-11.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The teacher edits an activity |
| **Test Case name:** | Edit activity |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**  **The teacher have already created an activity.**  **The user must be logged in.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects an activity. | System displays activity detils. |  |  |
| 2 | The user clicks edit. | System lets you edit previous inputs. |  |  |
| 3 | The user edits previous inputs. | The system validates for invalid input. |  |  |
| 4 | User clicks submit button. | System updates the activity. |  |  |
| Post-Condition:Activity is altered. | | | | |

**Table 42-Delete Activity**

|  |  |
| --- | --- |
| **Test Case:** | TC-12.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The teacher deletes an activity |
| **Test Case name:** | Delete activity |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**  **The teacher have already created an activity.**  **The user must be logged in.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects an activity. | System displays activity details. |  |  |
| 2 | The user clicks delete. | System displays “all information about this activity will be lost, do you want to continue?” |  |  |
| 3 | User clicks ok button. | The system deletes theselected activity. |  |  |
| Post-Condition:Activity is deleted. | | | | |

**Table 43-Delete Class**

|  |  |
| --- | --- |
| **Test Case:** | TC-13.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The user deletes the class. |
| **Test Case name:** | Delete class |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:**  **The user must be logged in.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. |  |  |
| 2 | User selects a class. Clicks delete on option menu. | The system displays a message (“Are you sure you want to delete this class?”). |  |  |
| 3 | User clicks ok | The system deletes the class. |  |  |
| Post-Condition:The class’s files are deleted. | | | | |

**Table 44- Remove student**

|  |  |
| --- | --- |
| **Test Case:** | TC-14.0 |
| **System:** | JABA: Online tool for basic Java programming |
| **Designed By:** | Researcher |
| **Executed By:** | Teacher |
| **Description:** | The user removes a student on the class |
| **Test Case name:** | remove student |
| **Designed Date:** | October 2016 |
| **Executed Date:** | February 2017 |
| **Pre-condition:User must be logged in.** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. |  |  |
| 2 | The user clicks view students  . | The system displays all students registered on the class. |  |  |
| 3 | The user selects a student then clicks remove. | The system will display a message “Are you sure?”. |  |  |
| 4 | User clicks ok. | The system removes the student from the class table. |  |  |
| Post-Condition:The student have been removed. | | | | |

**Verification, Validation and Testing**

**Unit Testing**

The function of unit testing is to test the smallest testable part of the software, separate it to test and determine if it operates as expected. Each unit was tested before integrating the parts into modules.

**Integration Testing**

Integration testing tests integration of interfaces between components, interactions to different parts of the system such as interfaces between systems and integration of system with the hardware.

**System Testing**

System testing is the final testing that the researcher has done. The researcher’s task is to make sure that the system will be functional. This includes testing based on the requirements, test cases and functionalities of the system. System testing is relevant to ensure the system delivers and meets the objectives and purpose of the study.

**CHAPTER 4**

**RESULTS AND DISCUSSION**

This chapter presents the results and discussion from the methods used by the researchers in the previous chapter.

**Project Description**

Jaba: Online tool for basic Java programming is a web based alternative learning system. The system helps to have an online way of posting activities, checking of their student’s codes and grading their works. The system is capable of creating activities for students as well as recording their grades. The system is based on an online learning website that students are already familiar in. In addition, the system provides access to some important features such as:

1. Online repository
2. Submit and resubmit an activity
3. Record graded activities
4. Create class
5. Export grades
6. Built-in compiler

**Project Structure**

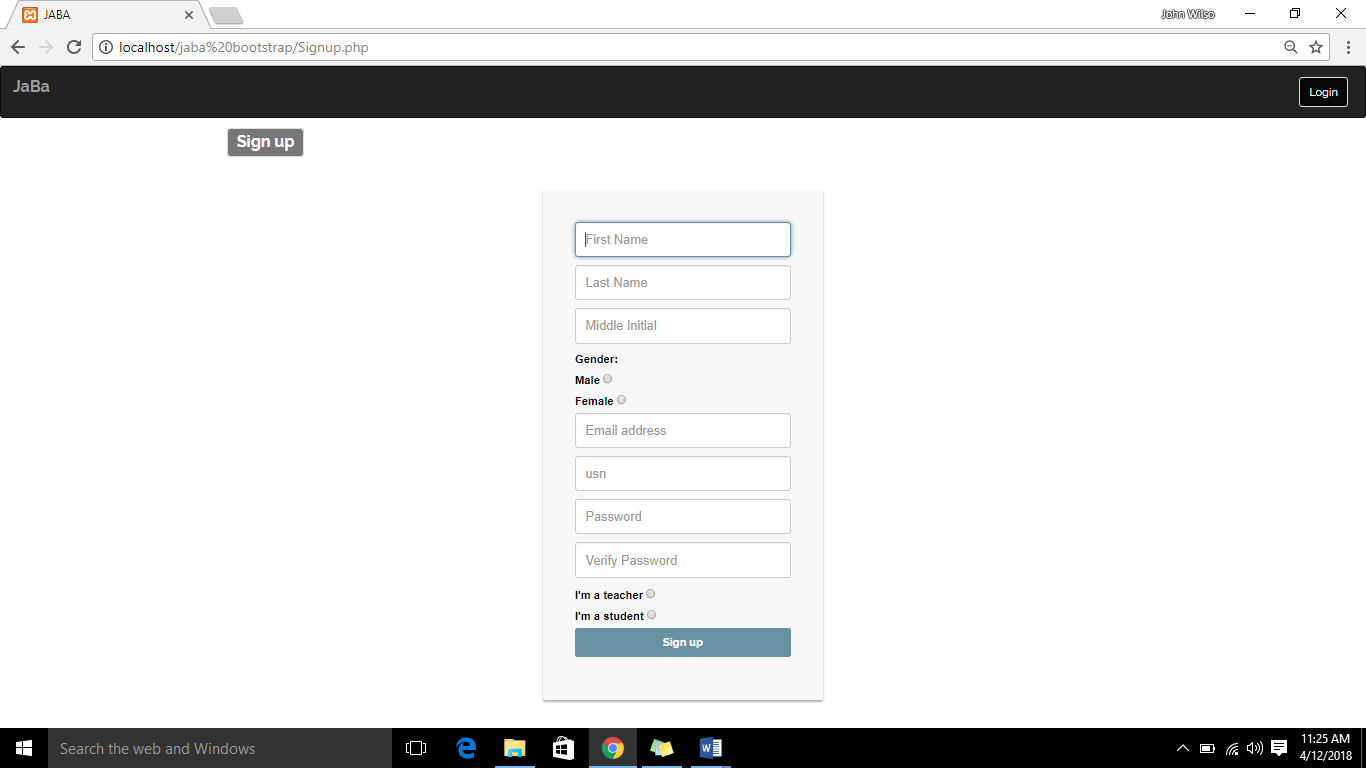


Figure 22- **Register**

Figure 22 shows the registration form of the website.

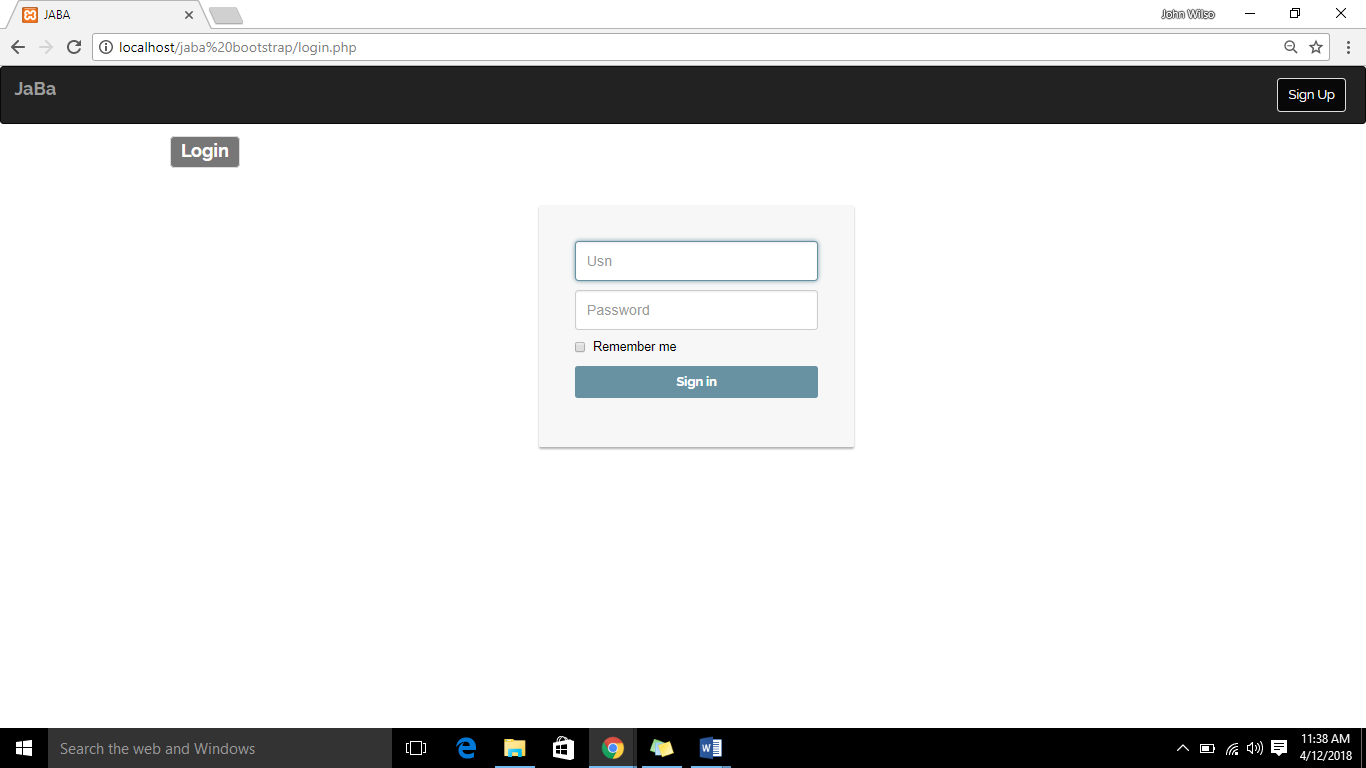


Figure 23- **Login**

Figure 23 provides the login form needed to access the website.

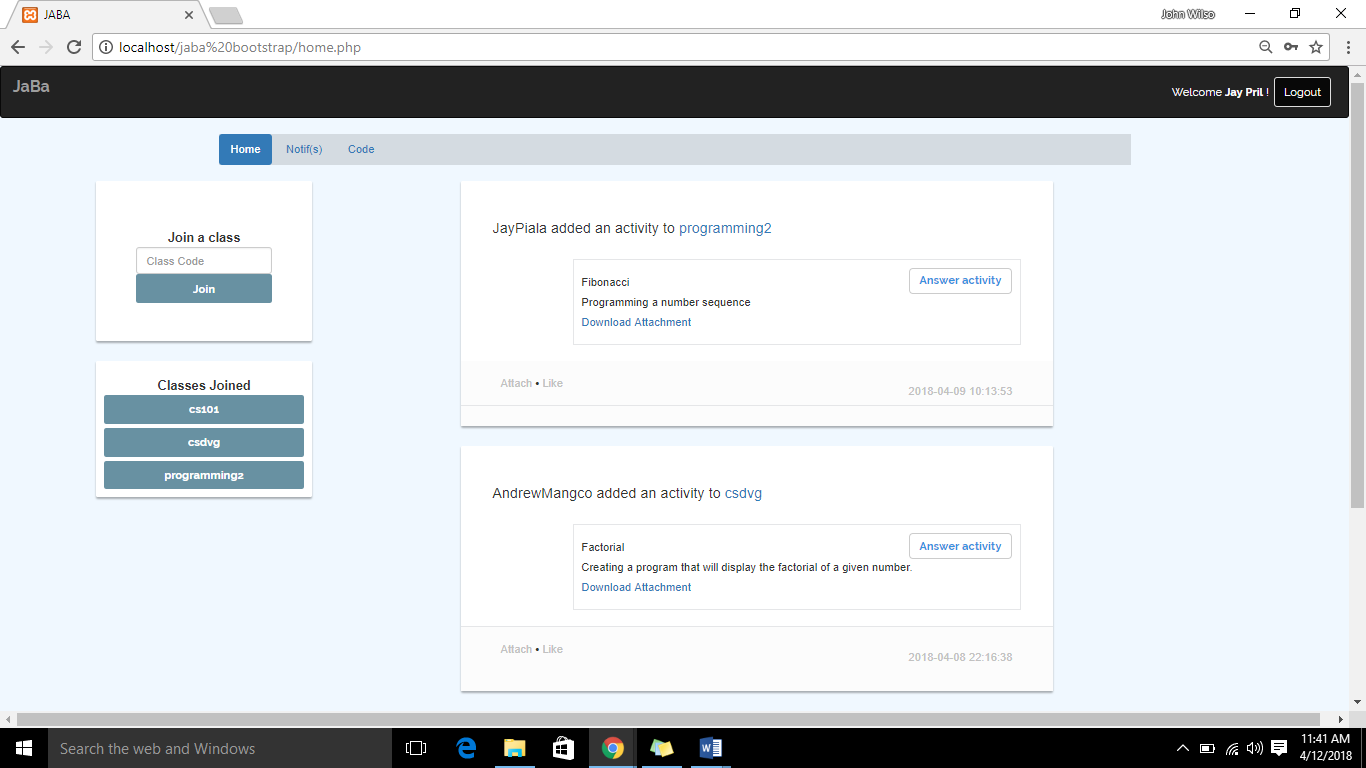


Figure 24- **Home(Student)**

The system’s home screen can be seen on figure 24.

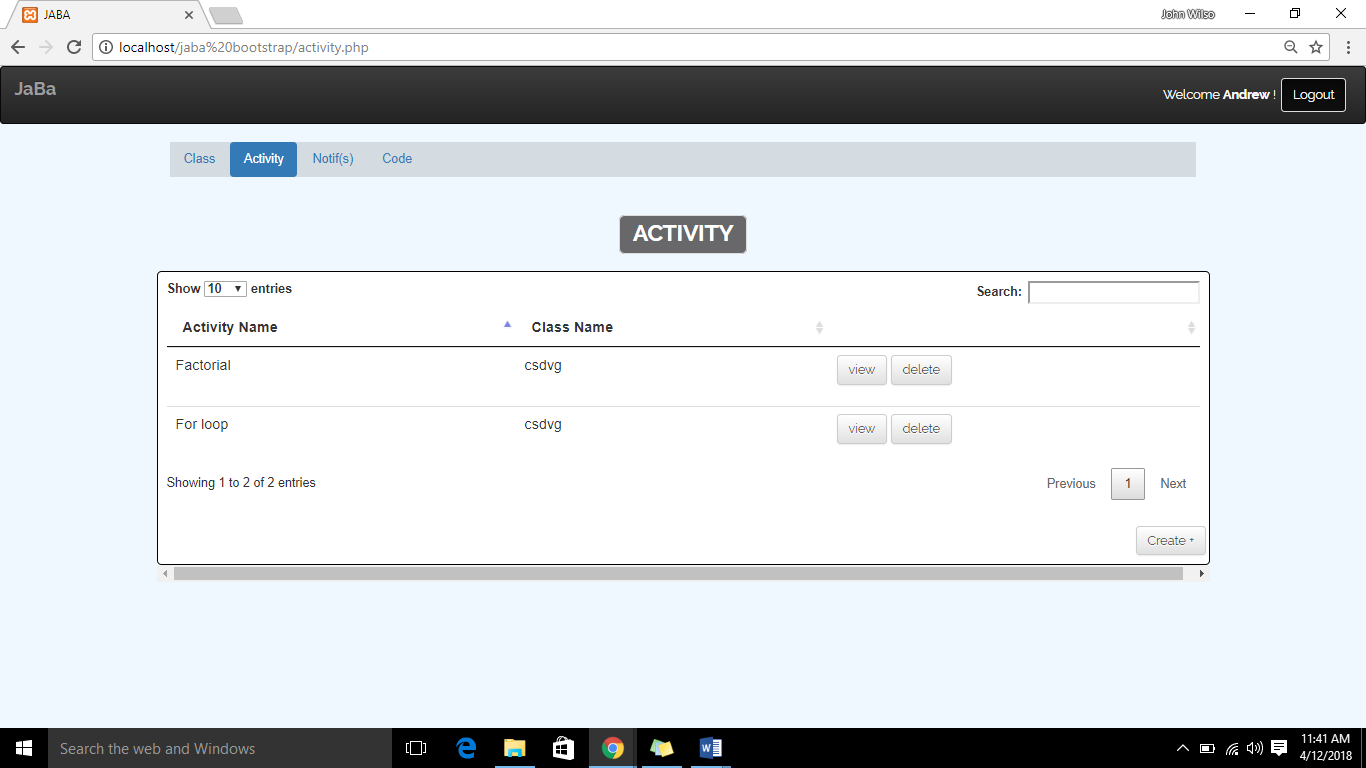


Figure 25**- Activity(Teacher)**

Figure 25 shows all the activities made by the teacher.

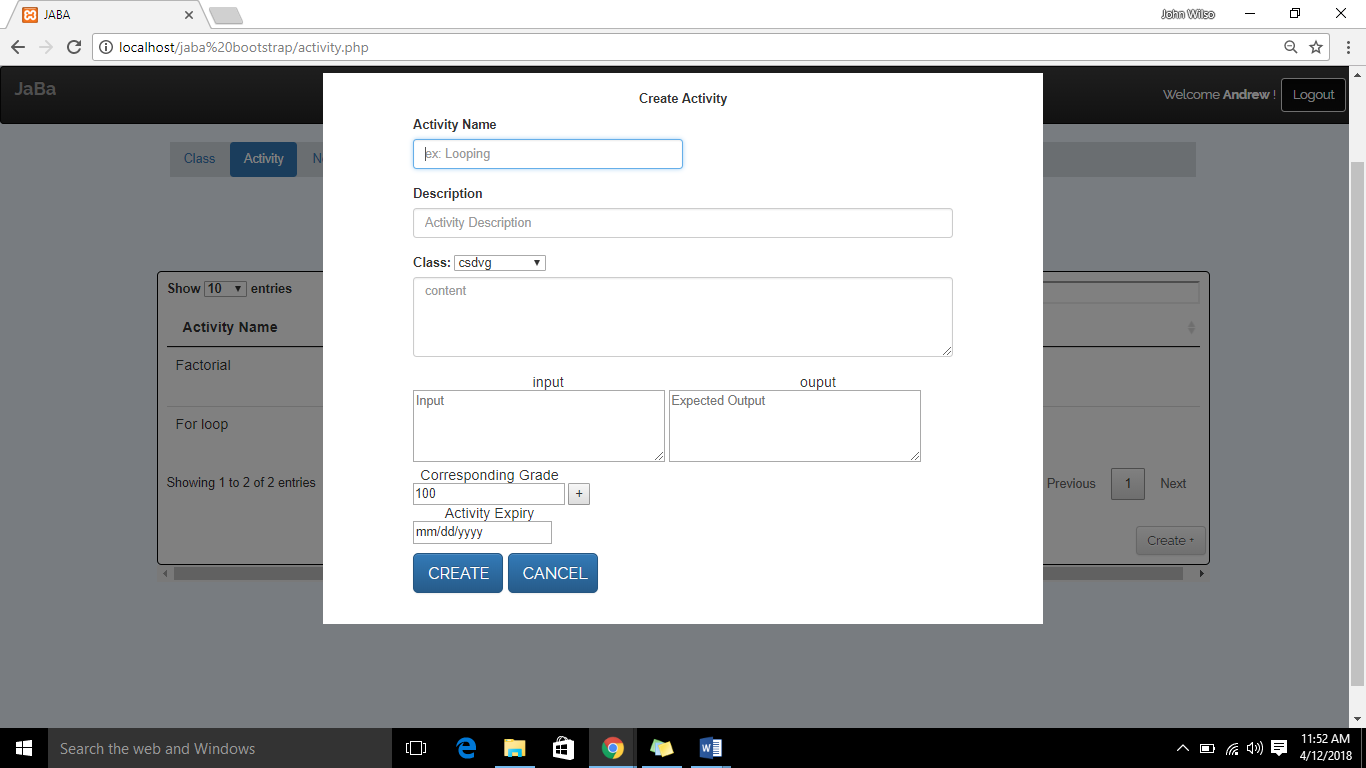


Figure 26- **Create activity(Teacher)**

Figure 26 provides the create activity form.

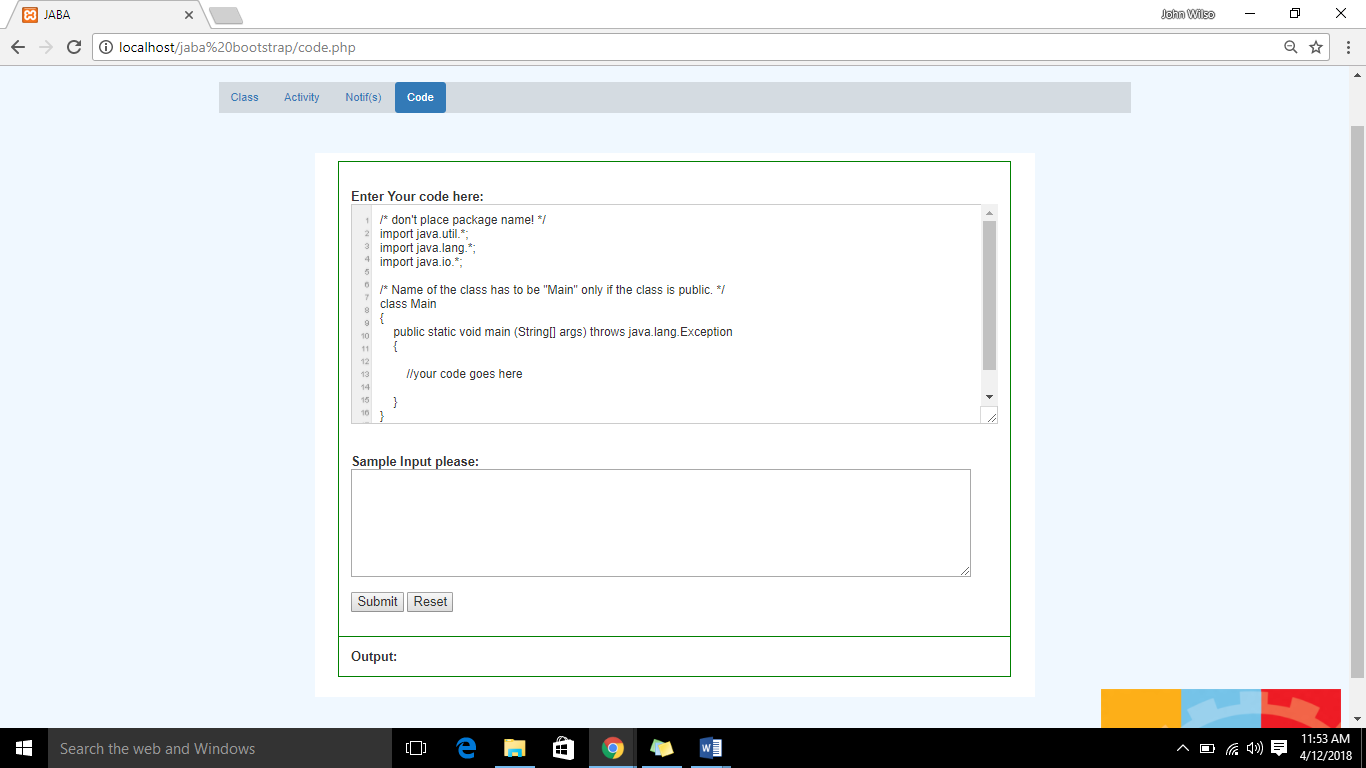


Figure 27**- Test Code**

Code testing page can be seen on figure 27.

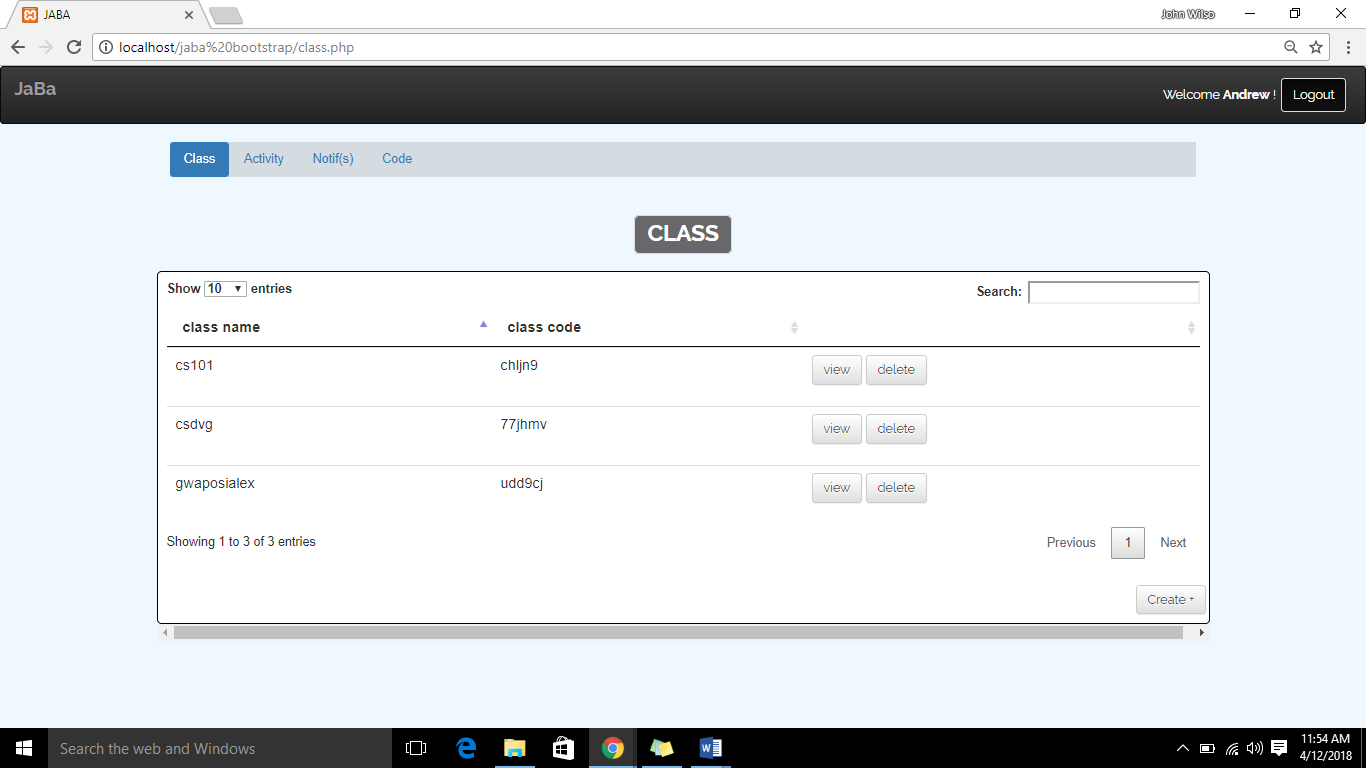


Figure 28**-Class(Teacher)**

List of classes created can be viewed on figure 28.

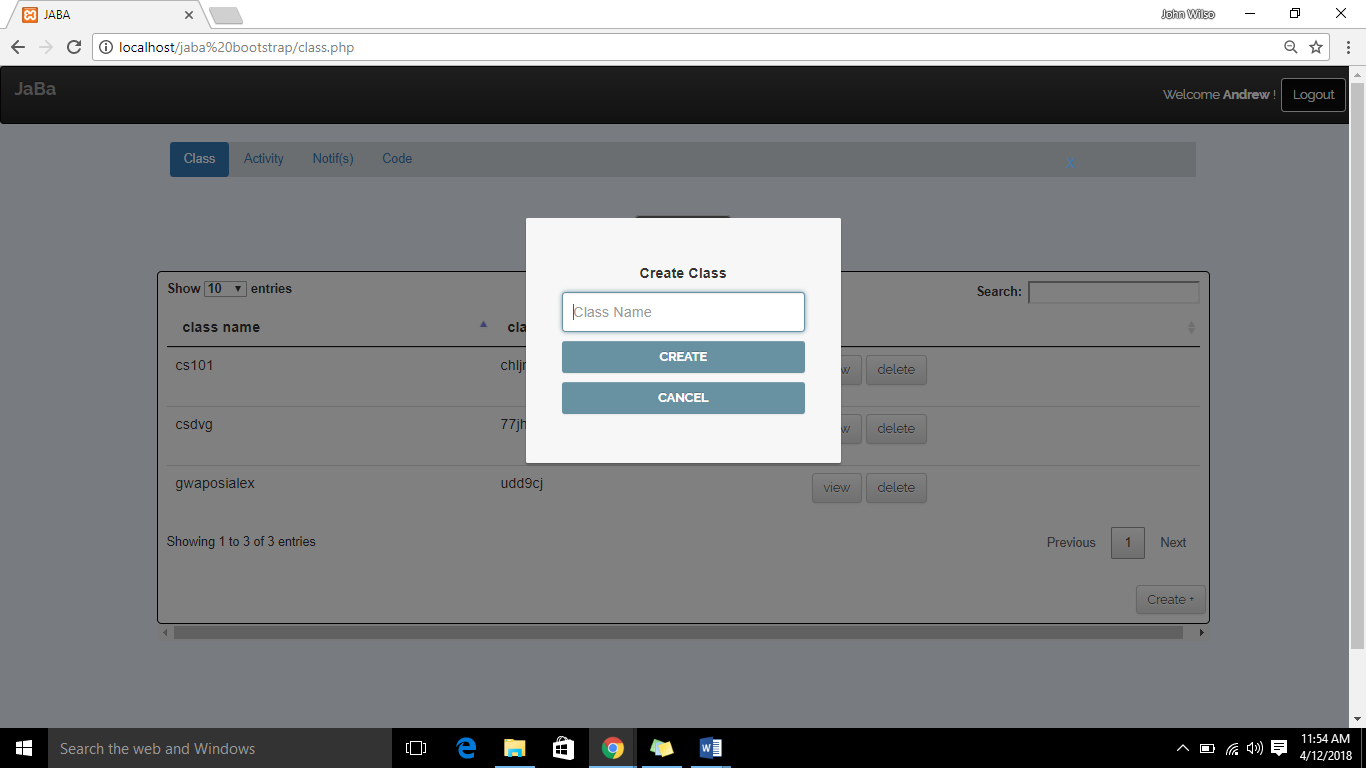


Figure 29**- Create Class**

Figure 29 shows how a teacher could create a class.

**Project**

**and Limitations**

As the system continues to grow, the needs and additional features also rise because of this problem; here are some capabilities and limitations of what the system can do.

**Capabilities:**

1. The system provides an online repository for created activities.
2. The system allows students to submit and resubmit an activity.
3. The system records graded activities.
4. The system allows teachers to create, edit, and/or delete class.
5. The system provides a list of the students’ grade for export.
6. The system provides an embedded compiler.

**Limitations:**

The system cannot control the students time of submission and if they prefer not to submit.

The system does not show notifications when an activity expires.

**Project Test Results**

**Test Case**

Test Results No: 1

Test Results Name: Login

Table 45 Test Results (Login)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | User click sign in button | Sign in form appears. | Pass |  |
| 2 | Input username and Password | System validates username and password. | Pass |  |
| 3 | User click submit | Website will change its status to logged. | Pass |  |
| Post-Condition: Website will be available for access. | | | | |

Test Results No: 2

Test Results Name: Register

Table 46 Test Results (Register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | Click register button. | The system will registration form appears. | Pass |  |
| 2 | Inputs required information | The system checks for invalid inputs. | Pass |  |
| 3 | Click submit | System displays “Registered successfully!” | Pass |  |
| Post-Condition: User may now login. | | | | |

Test Results No: 3

Test Results Name: Create class

Table 47 Test Results (Create class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | User click sign in button | Sign in form appears. | Pass |  |
| 2 | Input class name | System validates the class name. | Pass |  |
| 3 | User click submit | A message box appears with the class code. | Pass |  |
| Post-Condition:   1. A class is created. 2. A code is given to the user for the students to access the class. 3. The students may now join the class. | | | | |

Test Results No: 4

Test Results Name: Create activity

Table 48 Test Results (Create activity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | Click create activity button | System redirects to create activity form. | Pass |  |
| 2 | The user inputs activity information | System makes sure the input and output of the quiz is filled. | Pass |  |
| 3 | Click Create | A new activity is created successfully. | Pass |  |
| Post-Condition:   1. A class is created. 2. A code is given to the user for the students to access the class. 3. The students may now join the class. | | | | |

Test Results No: 5

Test Results Name: Submit activity

Table 49 Test Results (Submit activity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | click activity button | The website redirects to the answer activity page. | Pass |  |
| 2 | The user types the answer on the online compiler | The system compiles source code if answer matches the expected input and output. | Pass |  |
| 3 | User clicks submit button. | The system saves the grade (Pass/Fail) of the student. | Pass |  |
| Post-Condition: The activity has been submitted. | | | | |

Test Results No: 6

Test Results Name: View grade

Table 50 Test Results (View grade)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks grades on the navigation bar. | The system redirects to the grade page. | Pass |  |
| 2 | The user selects a class. | The system will display the tabulated list of grades. | Pass |  |
| 3 | User clicks export file button. | The system exports the graded table to local computer. | Pass |  |

Test Results No: 7

Test Results Name: Join class

Table 51 Test Results (Join class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user inputs the code to search for the class. | System searches for the code’s corresponding class. | Pass |  |
| 2 | The user clicks join class button. | A dialog box appears (“You are about to join a class.”). | Pass |  |
| 3 | User clicks ok | System adds the student to the class. | Pass |  |
| Post-Condition:The user successfully joined the class. | | | | |

Test Results No: 8

Test Results Name: Add student

Table 52 Test Results (Add student)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects students to be a member of the class. | The system displays the students registered to the teacher’s previous classes. | Pass |  |
| 2 | The user clicks add button. | System updates the student table. | Pass |  |
| Post-Condition:Student successfully added to class. | | | | |

Test Results No: 9

Test Results Name: Disable class

Table 53 Test Results (Disable class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. | Pass |  |
| 2 | The user selects disable button on the option menu. | The system disables the class. | Pass |  |
| Post-Condition: Class is disabled. | | | | |

Test Results No: 10

Test Results Name: Rename class

Table 54 Test Results (Rename class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks edit class button. | The system checks if the desired name is available | Pass |  |
| 2 | User clicks submit button. | The system changes the name of the class. | Pass |  |
| Post-Condition: Class is renamed. | | | | |

Test Results No: 11

Test Results Name: Edit activity

Table 55 Test Results (Edit activity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects an activity. | System displays activity details. | Pass |  |
| 2 | The user clicks edit. | System lets you edit previous inputs. | Pass |  |
| 3 | The user edits previous inputs. | The system validates for invalid input. | Pass |  |
| 4 | User clicks submit button. | System updates the activity. | Pass |  |
| Post-Condition: Activity is altered. | | | | |

Test Results No: 12

Test Results Name: Delete activity

Table 56 Test Results (Delete activity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user selects an activity. | System displays activity details. | Pass |  |
| 2 | The user clicks delete. | System displays “all information about this activity will be lost, do you want to continue?” | Pass |  |
| 3 | User clicks ok button. | The system deletes the selected activity. | Pass |  |
| Post-Condition: Activity is deleted. | | | | |

Test Results No: 13

Test Results Name: Delete class

Table 57 Test Results (Delete Class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. | Pass |  |
| 2 | User selects a class. Clicks delete on option menu. | The system displays a message (“Are you sure you want to delete this class?”). | Pass |  |
| 3 | User clicks ok | The system deletes the class. | Pass |  |
| Post-Condition: The class’s files are deleted. | | | | |

Test Results No: 14

Test Results Name: Remove student

Table 58 Test Results (Remove student)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected system response** | **Pass/Fail** | **Comment** |
| 1 | The user clicks class on the navigation bar. | System displays all classes created by the user. | Pass |  |
| 2 | The user clicks view students  . | The system displays all students registered on the class. | Pass |  |
| 3 | The user selects a student then clicks remove. | The system will display a message “Are you sure?”. | Pass |  |
| 4 | User clicks ok. | The system removes the student from the class table. | Pass |  |
| Post-Condition:The student have been removed. | | | | |

**Project Evaluation**

**Implementation Results**

The following are the respondent’s evaluation on the functionality, usability, content and knowledge of the AMACC-Davao students in Jaba: Online tool for basic Java programming.

**Table 56 Indicators**

|  |  |  |
| --- | --- | --- |
| **Indicators** |  | Interpretation |
| 1. The system allows creation of activities for the students to answer. | 4.53 | Strongly Agree |
| 2. The system allows to use the activities from previous classes. | 4.5 | Strongly Agree |
| 3. Activities created display accurate time. | 4.46 | Strongly Agree |
| 4. Checking laboratory exercises can be executed faster. | 4.5 | Strongly Agree |
| 5. System provides access of the students’ grades. | 4.6 | Strongly Agree |
| 6. System provides reading available grades for export. | 4.5 | Strongly Agree |
| 7. System allows access to a Java compiler. | 4.5 | Strongly Agree |
| 8. The website has a user-friendly interface. | 4.53 | Strongly Agree |
| 9. The system allows access to course activities. | 4.43 | Strongly Agree |
| 10. The system allows access to reference materials. | 4.63 | Strongly Agree |
| 11. The system provides convenience in submitting activities. | 4.63 | Strongly Agree |
| 12. The system gives notification on given activities. | 4.63 | Strongly Agree |
| 13. It allows resubmission an activity if needed. | 4.63 | Strongly Agree |
| 14. The system gives access to a Java compiler. | 4.63 | Strongly Agree |
| **Overall** | 4.5 | Strongly Agree |

The respondents’ evaluation on the features of Jaba: Online tool for basic Java programming for AMACC Students determines that the system meets its system’s objectives. Results show of 4.46 to 4.63 which have a verbal equivalent of Strongly Agree, which means that the system completed and satisfied the teachers and students expectations on an online learning tool.

**CHAPTER 5**

**CONCLUSIONS AND RECOMMENDATIONS**

**Conclusions**

Based on the respondent’s response, the researcher concluded that:

1. The system saves the activities and lessons of the students in an online repository.
2. The system allows teachers to check laboratory exercises faster.
3. The system facilitates grading and recording of laboratory exercises online.
4. The system provides Java compiler for easy access.

**Recommendations**

For the future development of the system, the researcher recommends the following features and functions that may help the system to be more reliable and useful to its users.

1. Improve UI design - The system needs to be trendy, the UI design of the system should be updated yet friendly.
2. Online one on one chat - other means of communication is always accepted. Chat is the most commonly used communication online that can be useful in this system.
3. Improving the compiler – Giving the compiler some upgrades would benefit users such as a debugger, error finder, etc.