Internship report on:

Online academic information Hub (OAIHUB)

Submitted by:

Shantonu Saha

ID:181-16-269

Spring 2020

Department of Computing and Information System (CIS)

Supervised by:

Nayeema Rahman

Senior Lecturer, Department of Computing and Information System (CIS)

Daffodil international university

Submission Date:22.06.20

©Daffodil International University

APPROVAL

This Project title **"Online academic information hub (OAIHUB)"**, Submitted by **Shantonu Saha**, ID No: **181-16-269** to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on 19-07-2020.

BOARD OF EXAMINERS

Mr. Md Sarwar Hossain Mollah **Chairman**

Assistant Professor and Head

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University

Ms. Nayeema Rahman **Internal Examiner**

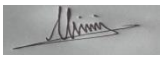
Sr. Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University

©Daffodil International University



Mr. Minhaj Hosen **Internal Examiner**

Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University

Dr. Saifuddin Md. Tareeq **External Examiner**

Professor

Department of Computer Science and Engineering

Dhaka University, Dhaka

©Daffodil International University

Acknowledgement

At First, honor and thanks to the Almighty god for His gifts of blessings during our study work for successfully complete the project.

*Then I like to say thank you to my Project Supervisor**, Nayeema Rahman**, Senior lecturer, Daffodil International University for providing me an amazing guideline. the conversation we had with him on research work and project planning, we extend our heartfelt thanks to my parents and other family members for their support, and patience. I am very grateful to our families for their devotion, prayers, consideration, and sacrifices to train and prepare us for our future. We are very much grateful to our other teachers and teammates for their devotion, understanding and continued support in completing this internship work.*

©Daffodil International University

Executive Summary

My six-month internship program work was with Daffodil international academy. I was involved in this as a software developer intern with their intern team name “DIAINTERNTEAM”. This report will cover some background information on the projects was involved in as well as details on how the project was developed. The report also states that which academic courses and projects helped me in overall in internship experience so far.

At the very beginning of the internship work I prepared several learning goals, which I wanted to learn about:

• How to explore working skills in a professional environment.

• How to utilize my gained skills and knowledge.

• How to acquire knowledge about software development life cycle.

• How to acquire knowledge about the development methodologies.

• How to gain fieldwork experience/collect data in an environment unknown for me.

• How to improve my interpersonal and technical skills.

• How to create network with professionals in the industry.

Soo many projects they are developing at that time when I was joined there as an intern. I have worked in OAIHUB (online academic information HUB) which is one of their major projects and I had a significant role in this project.my task was to do following:

©Daffodil International University

- Understand and working with spring boot framework.
- Understand and working with Thyme leaf.
- Understanding Rest API.
- Understand and working with GIT.

I obtain so many new technical skills though my work. I acquire new knowledge in front end development using Thymeleaf. I also brushed my HTML5, CSS3, BOOTSTRAP3, JavaScript, java skills while working there. This internship work also helps me to develop research and analysis skills. That work helps me to enrich my code documentation knowledge too. This report shows advantages of using spring boot framework and my working capabilities and detailed overview of that project where I was involved.

©Daffodil International University

Contents

©Daffodil International University

©Daffodil International University

©Daffodil International University

List of figures:

©Daffodil International University

Chapter 1 – Introduction

1.1 Description:

At present, people are usually depending on modern technology for their daily activities. But the education system in Bangladesh has not been digitized with modern technology yet. Education is one of the most important parts where we need to improve ourselves. There is a lot of gaps in our education system. There is no site where people can get the initial information for their children's admission or anything else. People are suffering many problems for the information gap from the institution. The guardians and students don't know about the procedure for their admission to any school, college or university.

So, we want to develop a system where students can get their necessary educational information. The system will contain all the information of all the educational institutions from school to university. They can know about the cost of the individual school, college or university and the procedure for admission to the specific institute. They can know about the facility of the institute, ranking of their desired institute. Each and everything information will be uploaded in this system so that students and guardians can be benefitted. The system will also contain all the public exam questions and answers. Students can share any questions and answer in this system and there are an admin and moderator who moderate the user activity, give them access to share questions or any other. There is a forum where they can discuss the questions or any queries. Any abusive post will be moderated by the moderator. Students can get any update information by this system. So, this system will be helpful for the students.

©Daffodil International University

Chapter 2 – Initial Study

2.1 Background of the project

In our country admission coaching business plays a significant role when a time periods comes to our students to get admission in school, college and university. They are facing so many problems at that moment. At first, they are suffering from information lacking about that institutions. So many students are unable to collect information from their preferred institutions by visiting that institutions. Sometimes they missed their admission test due to lack of valid information. Students can't decide that moment which educational institute is good for them. Most students don't know about payment scheme of an educational institutions. OAIHUB web application will capable to reduce all of these problems in future. It allows user to found all information from remote home.

2.2 Problem Area

Every year in our country students are facing so many problems when they are going to get admitted into a school, college, university, and national university. The student doesn't know which school, college, university and the national university is good for him. which documents are needed if they want to get admitted to their favorite institution? They also don't know about their payment system and payment amount. They also don't have any concept about their admission test exam questions and so many important information about their preferred institutions. Sometimes lack of proper information they missed their admission test exam.

©Daffodil International University

2.3 Possible solution

After analyzing all problems, I saw in our country students are facing so many problems when they are going to take admission to any educational institute. To reduce all these problems OAIHUB web application is the best possible solution. This system brings all academic information to its users. User also can discuss about their academic problems by creating post with other users. A user also can judge or provide a solution to a post via creating a comment. User can view previous admission test question of various year. They also can participate in those old admission test exams to improve their skills. User never miss any notification of any admission test what user wants to participate. In this system user also can view their educational institute rank and other important information also. OAIHUB also a great feature for pro users only which is a paid feature of this system. By using this feature user will get IELTS, GRE, SAT, etc. questions and answer for that questions and they also participate in online mock test exams.

©Daffodil International University

Chapter 3 – Literature Review

3.1 Discussion on problem domain based on published articles.

In our country people suffer from various kinds of problems in education sector.

- Most of the people does not know how to accommodate with the education system for their child.
- A father doesn't know which school will be suitable for his child and cost friendly for him and also well facilitated for both of them based on their situation.
- Sometimes people don't even know how much money to take out for admission fees.
- Sometimes some institution may show people that they're offering very affordable cost for people but later with time they demand too much high price for completion of their child's study. Which creates a heavy pressure on parents.
- When it comes to the question which college will be good and what type of study a student has to go through nobody knows the proper one.
- Nobody can answer what kind of specific preparation a student should take as there are thousands of coaching centers and book publishers offering their own methodologies which only leads to their own business purposes. Students are greatly suffered there.
- Students cannot find or have to buy previous question banks for high prices for taking preparation in the exams.
- Model test costs are very much high depending on coaching centers advertisements "Getting A+ in God Speed" or "Getting admitted into desired institution with zero study" which is not affordable for all of the students.

©Daffodil International University

- Students who have just graduated don’t know which companies to apply based on their skills.
- New graduates can’t give proper model tests for their specific job exams or interviews.
- Students cannot find answers of question banks, cannot take suggestion or teachings sometime to have the best answer for his problem.
- We do not have a discussion hub of our country for educational discussion or working purpose.

3.2 Discussion on problem solutions based on published articles.

Depending on all these problems various people came up with various solutions.

- Institutes started their own terms of marketing with benefits. Giving various offers to the students.
- A lot of mini coaching centers for school, college, university admission has created.
- Each and individual coaching centers had specified a specific publication of books to read.
- People started using various social media sites to find information about institutions and resources to study for giving exams.
- No specific Solutions has made to solve all the above-mentioned problems.
- Students with low lost budgets are missing thousands of chances and facilities to shine their life.
- With the digital technology people started learning accordingly how to cope up with all of this term by term.

©Daffodil International University

3.3 Comparison of three/four leading solutions-

Best features

- Digitization has made people's life easier. People can easily access to their required information though they have to surf for it too much.
- People can rely on specific institutes for getting admitted into them.
- Various information can be found on various places on internet about the institutions, course curriculum, cost and a lot more.
- Some question answer sheets of various years can be found on internet.

Limitations

- Information are lack of accuracy. Proper information cannot be found.
- Different sources tell different information. Questions and answers are not found with accurate guideline or answers.
- Job applications or advertisements are not accurate.
- People do not have the ability to speak with the specialists for better solutions.

3.4 Recommended approach

- ✓ All the educational information and related things should be brought together.
- ✓ All students should be treated equally.
- ✓ Parents should know what they are doing. Where their children are getting admitted, is it affordable and maintaining for the parents.
- ✓ Students must have the ability to decide where they want to study and grow their future career.
- ✓ All educational equipment should be very much affordable so that no one misses their rights.

©Daffodil International University

- ✓ Students can give their model test for very much affordable cost for getting prepared for the exams or jobs.
- ✓ A place where all the legal information will be found about each and every educational institution to make decisions for the children. Where People can compere between the institutions and decide which will be better for their children and affordable for the parents.
- ✓ Nothing should be compromised when it's the question of education and the future of our country.
- ✓ Whenever any student is asking a question or in a problem will have the ability to share it in somewhere where all kinds of specialists will be available to give solutions.
- ✓ Nobody will miss their education rights to study and brighten their future.

©Daffodil International University

Chapter 4 – Methodology

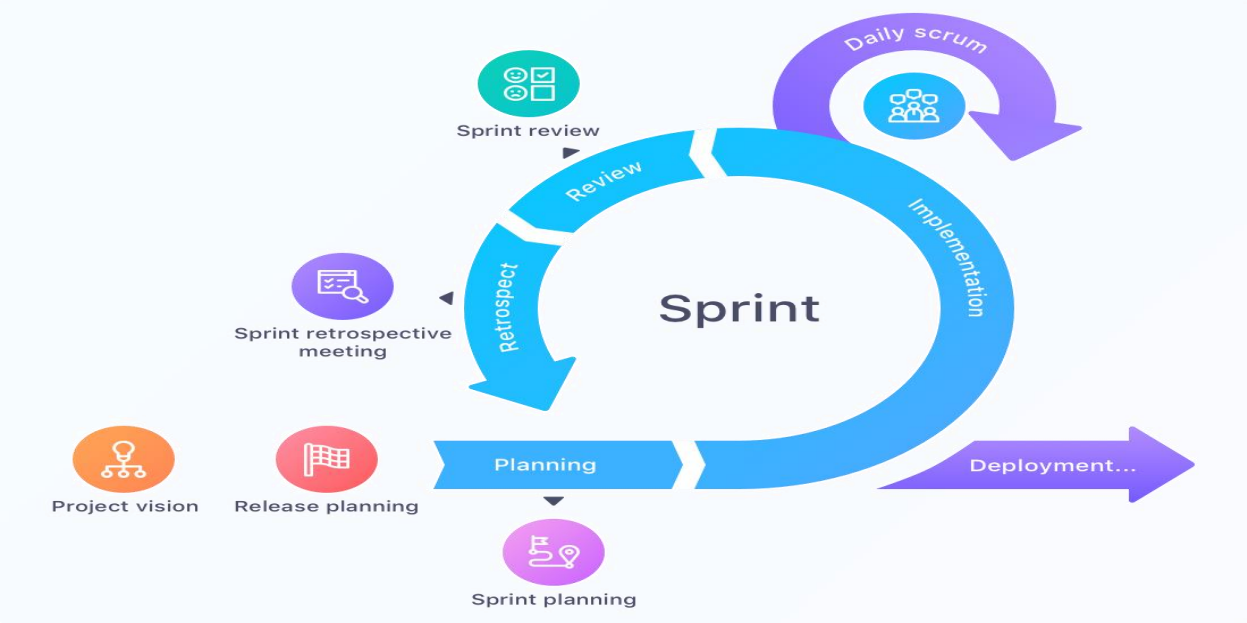
Methodology is a set of procedures or a particular procedure. It helps to provide appropriate guideline principle for developing an application or system.

4.1 What to use

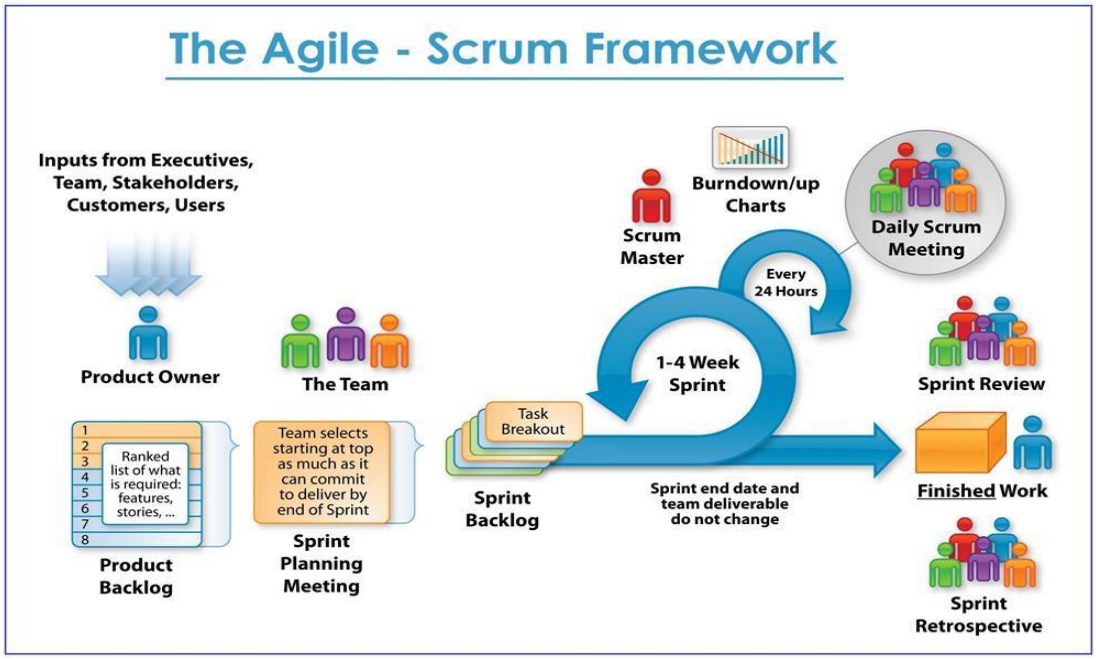
In software development site, there are so many methodologies for developing an application. Agile is one of them. Actually, agile is an evolutionary project management approach under which requirements and solution evolve through the collaborative effort of self-organizing/ cross-functional teams and their customer/end users.

it is a project management methodology what uses small development cycles name “sprints” to attention on continues improvement in the development of an application or a system.

©Daffodil International University



In this project development our team has been conducted with scrum framework of agile. This framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is lightweight, simple to understand and difficult to master.



©Daffodil International University

4.2 Why to use

The reason of using Scrum framework is given below:

- Higher productivity.
- Better-quality products.
- Reduced time to market.
- Improved stakeholder satisfaction.
- Better team dynamics.
- Happier employees.

4.3 Sections of methodology

There are three pillars of Scrum.

- Transparency.
- Inspection.
- Adaption.

©Daffodil International University

Transparency
Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

Inspection
Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

Adaptation
If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

4.4 Implementation plans

Agile implementation is a form of project management that works in small increments and well suited to projects that could be become irreverent once delivered, especially useful in software development. The key to the agile plan is that it provides flexibility for changes to the product as it continues to be developed. Scrum is a framework of agile what delivering product iteratively and incrementally in a timebox fashion. This is simple illustration of what the scrum implementors and others define it, moving with it.

©Daffodil International University

Chapter 5 – Planning

Here I'm going to show the entire internship work planning in a way that the internship work is being done by me. The whole work is divided in small pieces and those are done within the fixed period of time. In this phase a specific task when will be started and when will be end those things are defined.

5.1 Work breakdown structure:

WBS (work breakdown structure) is a tool what make the work more manageable and approachable. This approach helps to complete all the task within fixed time duration. The WBS (work breakdown structure) chart of my whole internship work is given below:

SL	Task title	Start date	End date	Durations (Days)
1	Introductions	01.01.20	02.01.20	2
2	Initial study	03.01.20	09.01.20	7
3	Literature review	10.01.20	14.01.20	5
4	Methodology	15.01.20	18.01.20	4
5	Planning	19.01.20	24.01.20	5
6	Foundation	25.01.20	30.01.20	6

©Daffodil International University

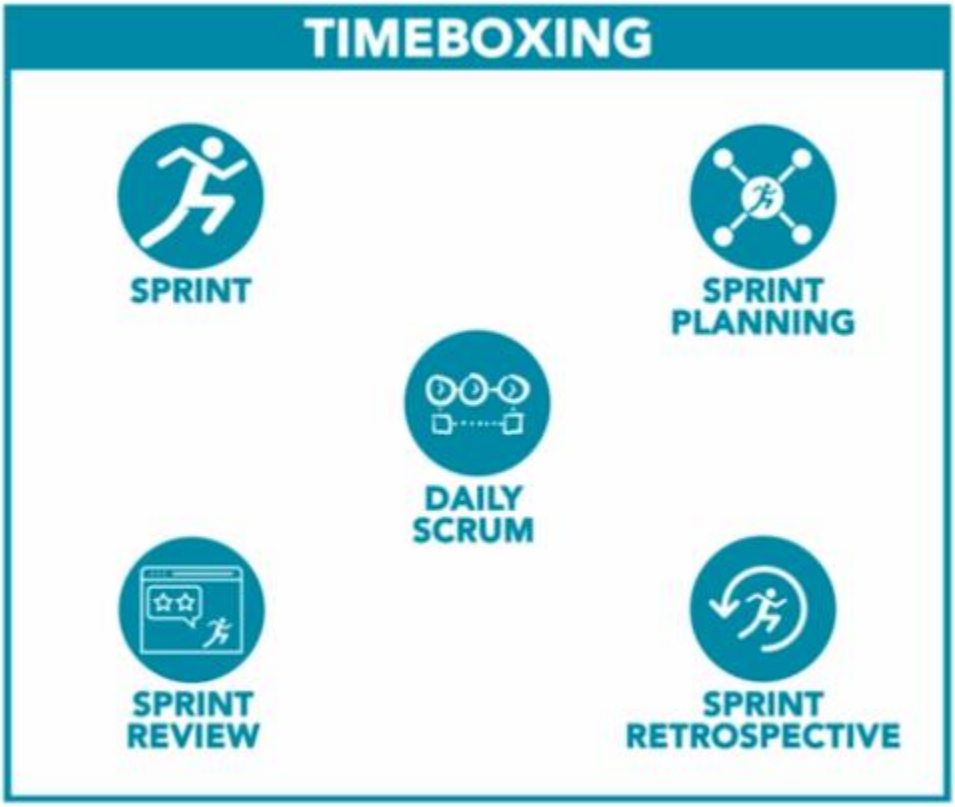
7	Exploration	01.02.20	04.02.20	5
8	Engineering	05.02.20	09.02.20	5
9	Deployment	10.02.20	01.03.20	20
10	Testing	02.03.20	11.03.20	12
11	Implementation	12.03.20	21.03.20	12
12	Critical Appraisal and Evaluation	22.03.20	28.03.20	7
13	Conclusion	29.03.20	30.03.20	2
Totals				**90 days**

Figure 1:WBS chart

©Daffodil International University

5.2 Time duration and time boxing:

The whole system is being done with agile methodology. All the tasks are iterative in this approach. Timeboxing is one of the useful parts for this project management approach (Anon., 2019). In this timeboxing process that we follow shown here below:



This is one of the critical components of a good scrum. **Sprint** used for utilize the length of the project. **Sprint planning** is actually a meeting what timeboxed 8 hour or less for a one month. **Daily scrum** process is actually a timebox for 15 minutes per day what helps to synchronize teams' activities. **Sprint review** also a time boxing to adapt the backlog based on feedback. **Sprint retrospectives** is an event what inspect itself identifies processes.

©Daffodil International University

5.3 Gantt chart:

This chart shows activity schedule of the projects and the time duration of specific tasks of the projects. This thing is very important for developing a project. Gantt chart of this projects is given below:

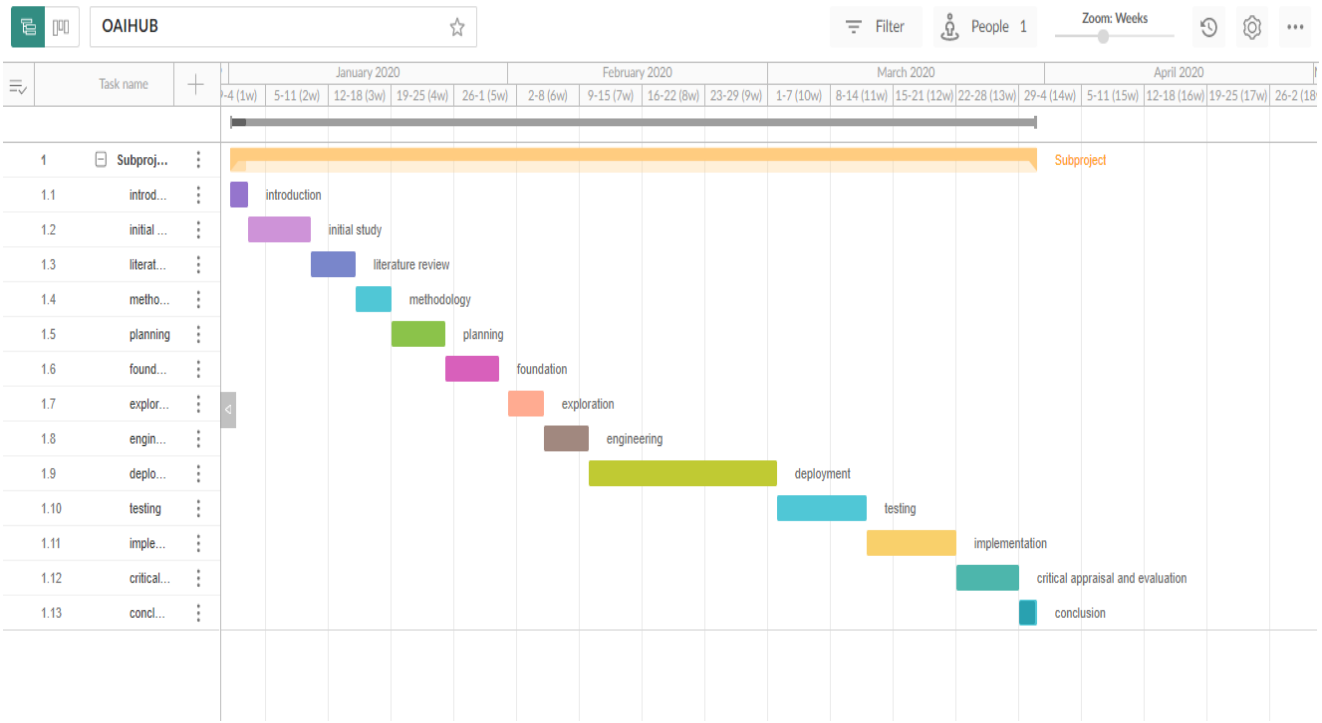


Figure 2:Gantt chart.

5.4 Test plan:

A test plan is a detailed document that describes the test strategy, objectives, schedule, estimation and deliverables and resources required for testing (Anon., 2020).

Required test:

Basically, there are two types of major testing. They are functional and non-functional. Those testing procedures are given below:

©Daffodil International University

Functional testing:

Unit testing: in this stage every single unit of a software are being checked. The main purpose of unit testing is that it can performs it is designed.it also increases maintainability of the codes. (Anon., 2020)

Integration testing:

In this testing stage a group of units is being tested in a single time. If all units perform together successfully then it will consider as a successful test.

Non-functional testing:

Security testing: this test defines how much secure a software is.it rescues web application from vulnerabilities, threats, risks etc.

Reliability testing:

This test defines that how much reliable a software is. when the test is occurred user also involved there.

©Daffodil International University

5.5 Test case:

Test case actually a set of conditions what helps a software to meet client requirements. So many types of test case are used in software testing. Here I've have used one of them that things are given below:

Test: 1		Test Class:	Designed By	
Data Source:		Objective:	Tester:	
Test Case	Description	Tasks	Expected Result	Actual Result
1.1		:		

Figure 3:Test case.

©Daffodil International University

5.6 User acceptance test plan:

This testing procedure is the last stage of testing. This test defines that testing performance by the user who clarify the systems that requirements are fulfilled or not. Every software needs to pass this stage before going to market.

Test Priority:	**Test Execute by:**
Unit test No: 01	**Test Execute Date:**
Test case	
Objective:	
Data Source:	

Case No.	Description	Tasks	Result		Status (Pass/Fail)
			Actual result	Expected result	
1					

Figure 4:Test plan.

©Daffodil International University

Chapter 6 – Foundation

6.1 Overall Requirement List

To Build the preliminary system we need following things to be implemented for constructing the project. The basic requirements will probably able to cut out the edge of the project which has been planned out. The requirements which has been analyzed for so long to develop this system will going to be cover maximum objective requirements of the proposed system. Here is all the overall requirement list given below:

- ✓ To register & save new user
- ✓ To register & save new moderator
- ✓ To register & save new university moderator
- ✓ To give control of the whole system in one hand (Admin)
- ✓ To register & save pro user
- ✓ To upload and download files via specific users
- ✓ To upload verified question & answer sheet by moderators
- ✓ To Upload & view institute details
- ✓ Compare between institutes details
- ✓ View & edit user profile
- ✓ View & edit moderator profile
- ✓ View & edit pro user profile
- ✓ View & edit university moderator profile
- ✓ To register normal registered user as pro user through payment
- ✓ Pro user has access to most of the things
- ✓ Pro user can view & download question – answer sheets
- ✓ Pro user can give model tests
- ✓ Pro user can apply for model tests
- ✓ Pro user can apply for specific institutes online
- ✓ Education board's various types of exam routines will be shown in the notice board
- ✓ UGC notices will be shown through moderators.
- ✓ Institute's over all details like cost, facilities, study quality, admission details everything will be shown
- ✓ A discussion forum or system is needed
- ✓ Every user, requirement-based moderators, pro users, will be able to post the discussion topics as threads
- ✓ Other users can comment under those post.
- ✓ Important threads can be upvoted via rating system

©Daffodil International University

- ✓ Threads will be attached with tags to make it retable to specific topics
- ✓ Model tests will be examined and moderated via top notch teachers.
- ✓ Specific thread publisher names will be visual individually
- ✓ Specific comment publisher names will be viewed
- ✓ Every thread will be viewed by time and date.
- ✓ Comments under a thread will be viewed by date and time. Important file will be able to uploaded through users, pro users in the threads e.g. snapshots, code snippets, word docks, images, and many more.
- ✓ A strong secure database system is needed to store all these information part by part and sequentially
- ✓ All specific details will be analyzed and saved via the system.

6.2 What Technology to be implemented (Client/Web/Standalone)

The technologies and languages which are going to be implemented in this system to develop the proposed system are given below:

Technical Languages

Java, JavaScript, AJAX, XHTML, CSS, Json, JSP, JSTL, HTML, Codemix, NodeJS, Bootstrap, jQuery,

Databases Systems

MySQL, Tomcat, JDBC,

Technologies

ORM (Object Relational Model) tool, Hibernate, REST API, Restful API, Data JPA, Spring Boot, Spring Security, Spring Boot Dev tools,

Framework

Spring, Thyme Leaf

Build tool

Maven, Gradle

Server Platforms

Daffodil Web Server Storage

©Daffodil International University

Chapter 7 - Exploration

7.1 Old Full System Use Case

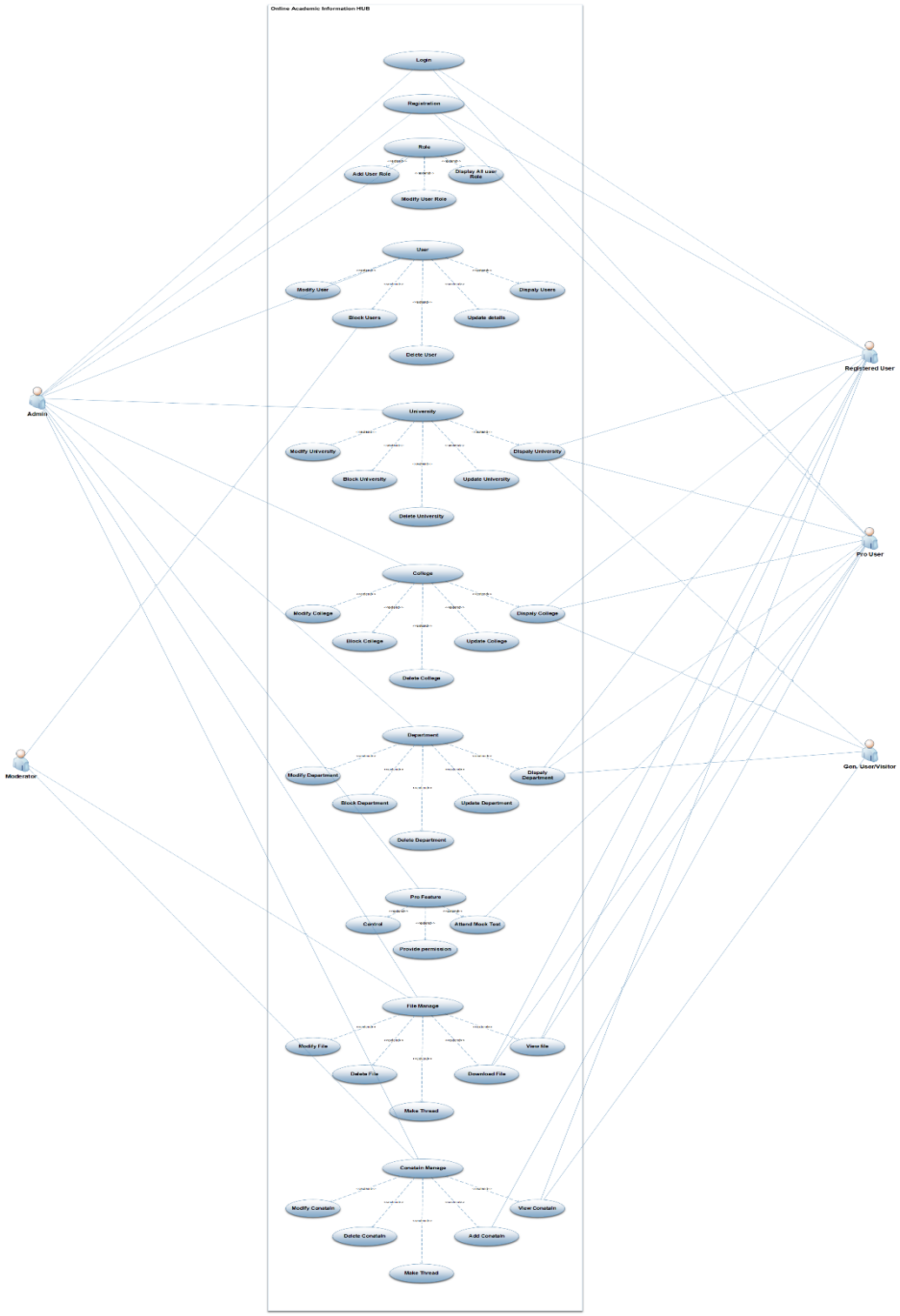


Figure 5:old full system usecase diagram.

©Daffodil International University

Description:

In this information hub there are total 5 types of user. Every user has their own rule to use this system. They have various type of data access in this system such as:

1. **Admin**: He/she can register themselves into the system and can login by their personal information. They can make and modify role in the system, can add user, display all user role. Modify user, block user, delete user, update user data those also can be accessible by admin. All type of University, College and Department related data can be display, blocked, modify or delete by the admin. They also have some pro feature such as control, provide permission and attend mock test (tester). All necessary file can be handle by the admin panel, modifying files, delete files, make thread, download file and view file are the admin panels task. Contain manage like modify, delete, make thread, add and view contain.

2. **Moderator**: This panel has less power and access then the admin. Modify user, block user, delete user, update user data those also can be accessible by moderator panel. Necessary file can be also handled by the moderator panel. Modifying files, delete files, make thread, download file and view file are the admin panels task. Contain manage like modify, delete, make thread, add and view contain.

3. **Register user**: This panel they can login and register in the beginning. They can see university, college and department data. From this panel they can also attend mock test. They will have access to view file and download them. They can able to add content and also view others.

©Daffodil International University

4. **Pro-user**: This panel members are the special then register member. This panel member can login and register in the beginning. They can view university, college and department all data. From this panel they can attend mock test. They will have access to view file and download them. They can also able to add content and also view others.

5. **General user/visitor**: They have the less ability in this system. Visitor can register themselves. Then they can do many things. Without registration they can only view can view university, college, department data and some content.

©Daffodil International University

7.2 Old Full System Activity Diagram

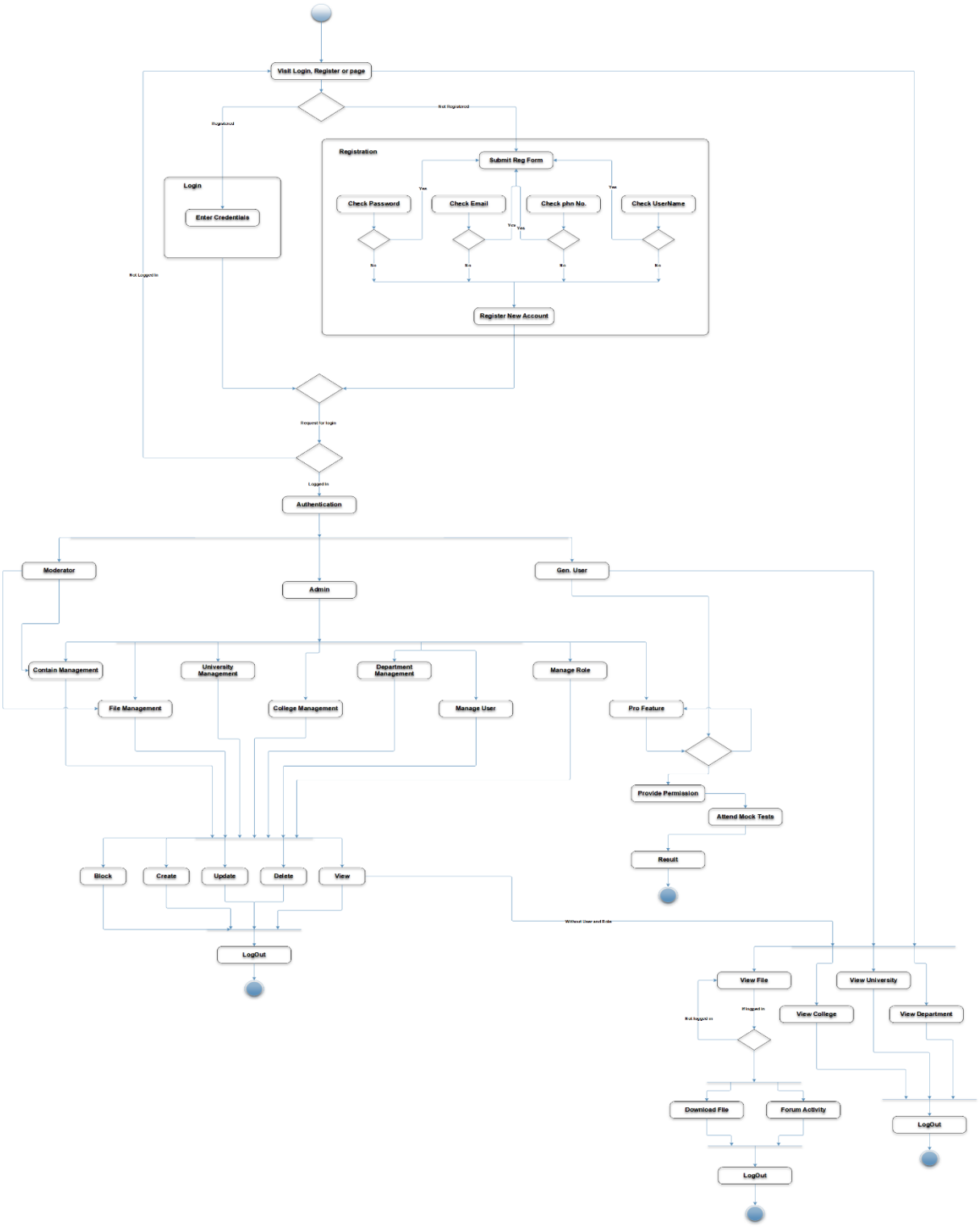


Figure 6:old full system activity diagram.

©Daffodil International University

Description:

In the activity diagram above the whole procedure that can be undertaken by a user are shown. If the user is registered, he will go the login page and log into the system providing valid user ID and password. If he is not registered, he will go to the registration or sign up page. He has to input some information like name, password and other relevant information. After signing up he will be able to go to the sign in page to enter the system. After signing in there will be three type of access. Those are admin, moderator and general user. Admin can create, delete, update and retrieve any data or account. Admin will also have to access to block any user. Moderator will have the access to the content management and file management. For general visitor there will be a view access where the user can only view the system and its contents. But if they want to download or download any content, they have to sign in there. After signing in they will have the access to pro features. In pro feature they can attend the mock tests from the question bank we have in our database. The admin will have the access to all the features like content, file management, University management, College management, department management, User management, role management and pro features. After using the system all kind of user will be able to log out from the system using a logout function.

©Daffodil International University

7.3 Prototype of new system

System architecture

Architectural design: MVC architecture

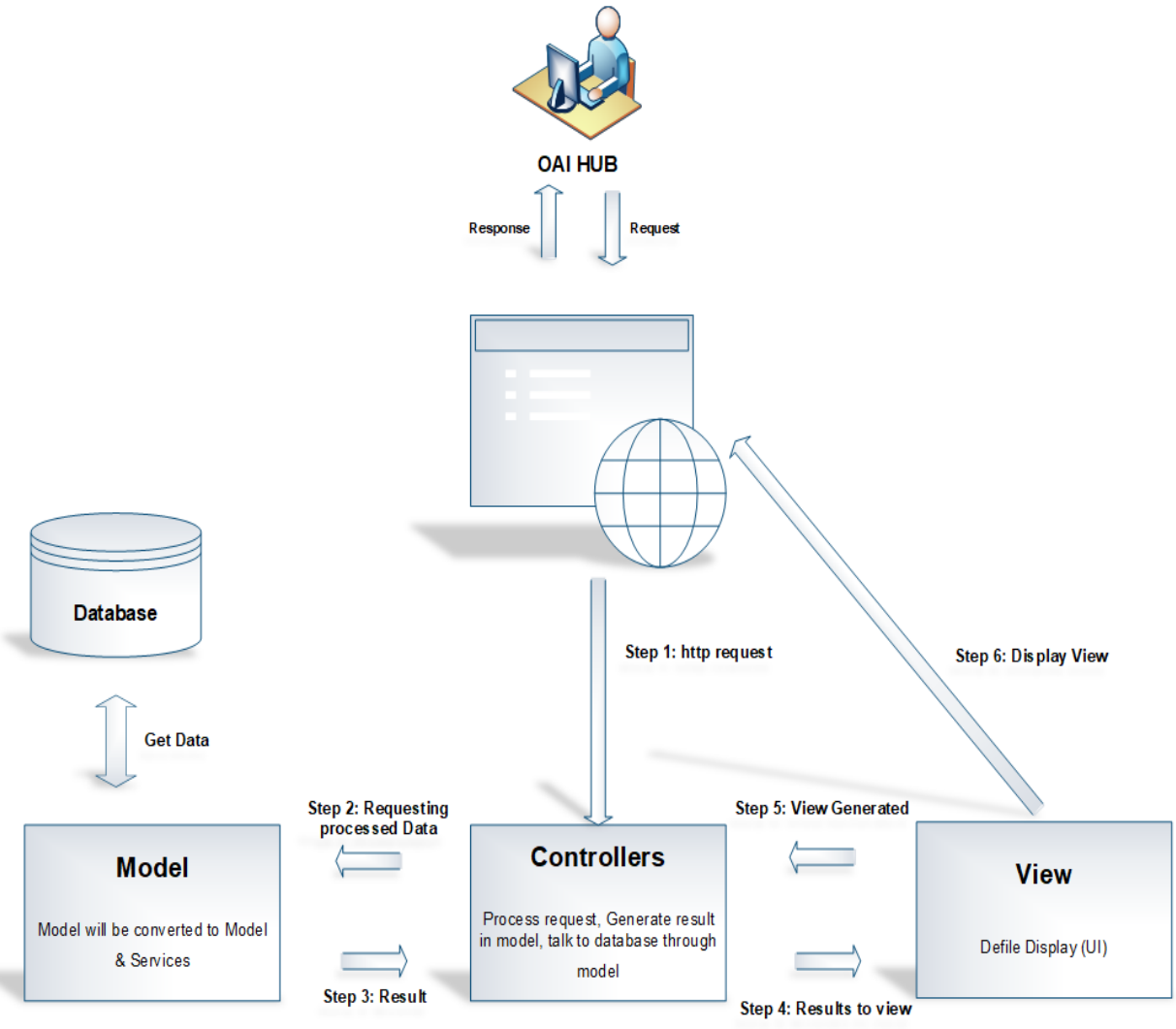


Figure 7:MVC architecture.

Description:

It's a system architecture diagram. We followed the design pattern of MVC for our system design. When a user make request our system through webpage then the webpage make a http request to the controller as a first step. Controller then process the http request and

©Daffodil International University

generate result in the model. Model gets data from the database. Here model is acting like a bridge between database and controller. Model does the definition and validation those data which age coming from the database. Then model send data to the controller. Controllers then pass those arrange data to view. View shows those data with a nice user interface through the modern webpage. User interface which is usually seen by the end user. That request and processing are the backend task. If any user make interaction with UI or make some input then the UI send those inputted data to the controller and controller send them to the model for analysis and arrange and check validation of those data then model send them to database to store those user data safely and securely. Then data base store those data and use shows on the UI that his or her data are securely store in the system database.

Chapter 8 - Engineering

8.1 New System Modules

In this Web Application The newly proposed and under developed modules are fascinating. A total discussion hub, the forum is going to be added and implemented in this system. Some pro features are going to be introduced for which user will have to pay to use. Forum discussion will be much like stack overflow. People can discuss about study, question answers, talk about institutions, post and share job articles and exams, give mock test, find all kinds of resources every student need in every exam, as it will also going to be connected with Education boards of Bangladesh & UGC. So, Students, users will find almost every facility to accommodate.

©Daffodil International University

8.2 Use Case

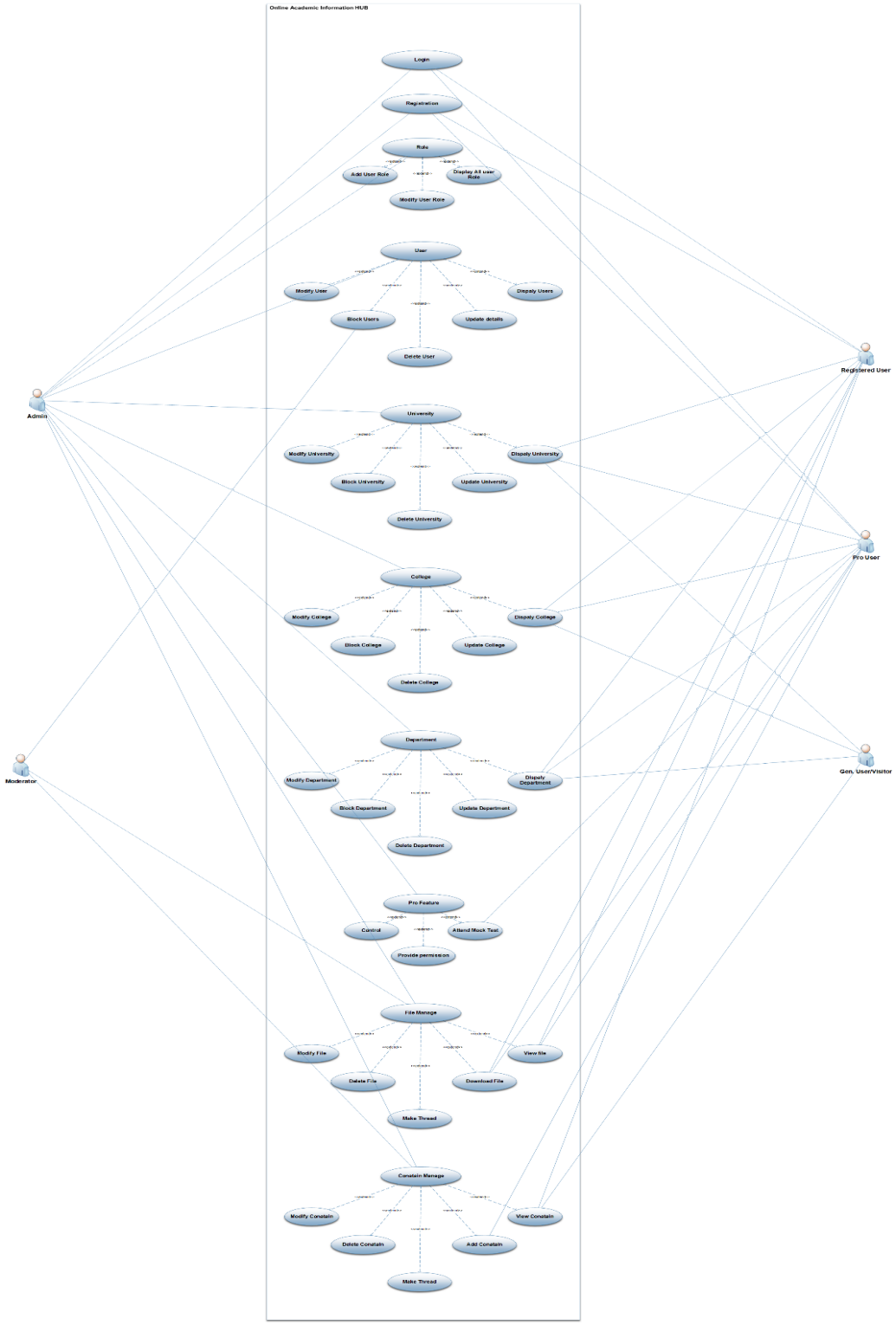


Figure 8:usecase.

©Daffodil International University

Description:

In this online academic information hub, there are total 5 types of user. Every user has their own rule to use this system. They have various type of data access in this system such as:

1. **Admin**: He/she can register themselves into the system and can login by their personal information. They can make and modify role in the system, can add user, display all user role. Modify user, block user, delete user, update user data those also can be accessible by admin. All type of University, College and Department related data can be display, blocked, modify or delete by the admin. They also have some pro feature such as control, provide permission and attend mock test (tester). All necessary file can be handle by the admin panel, modifying files, delete files, make thread, download file and view file are the admin panels task. Contain manage like modify, delete, make thread, add and view contain.

2. **Moderator**: This panel has less power and access then the admin. Modify user, block user, delete user, update user data those also can be accessible by moderator panel. Necessary file can be also handled by the moderator panel. Modifying files, delete files, make thread, download file and view file are the admin panels task. Contain manage like modify, delete, make thread, add and view contain.

3. **Register user**: This panel they can login and register in the beginning. They can see university, college and department data. From this panel they can also attend mock test. They will have access to view file and download them. They can able to add content and also view others.

©Daffodil International University

4. **Pro-user**: This panel members are the special then register member. This panel member can login and register in the beginning. They can view School, college, University admission test question papers and o levels, A levels, GRE, SAT etc. premium questions papers. From this panel they can attend mock test. They will have access to view file and download them. They can also able to add content and also view others.

5. **General user/visitor**: They have the less ability in this system. Visitor can register themselves. Then they can do many things. Without registration they can only view can view university, college, department data and some content.

©Daffodil International University

8.3 Pro-user use case:

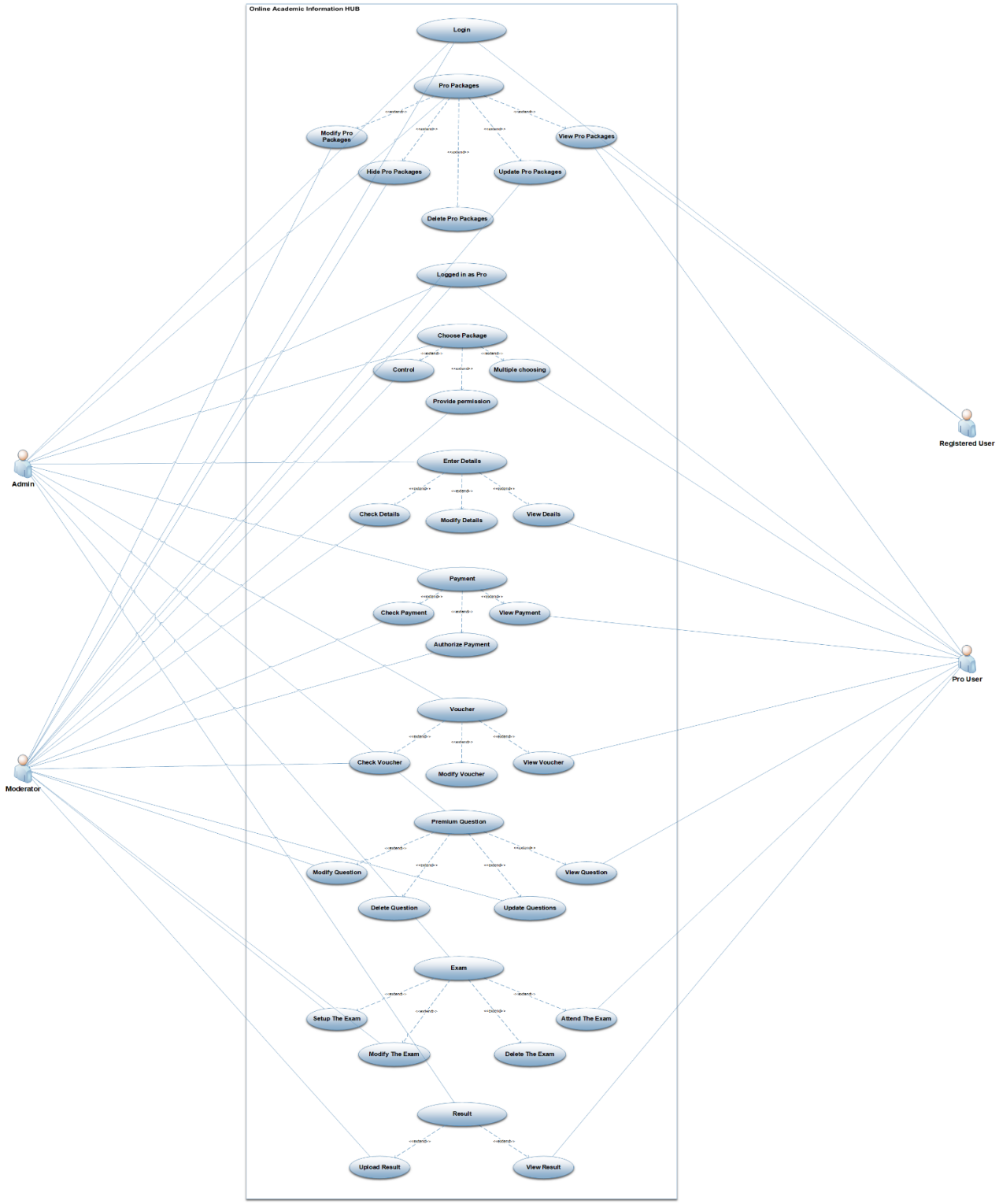


Figure 9:pro-user usecase diagram

©Daffodil International University

Description:

In this online academic information hub, there is different types of facilities for a pro user. Pro user has some different rule to use this system. They have various type of data access in this system such as:

At first registered every user in this system can view all the pro packages. The registered user who purchase any premium packages from this system they get a title of pro-user. Only admin has the facility to provide the title. Those pro packages bring premium questions bank and their solutions. This feature specially designed and developed for English medium students. They will get O levels, A levels, SAT, GRE etc. questions and their solutions from here. They also can participate in online Mock exams. All exam papers will be judged by a system moderator and the results will be shown in system when their exam will be finished.

©Daffodil International University

8.4 Class Diagram

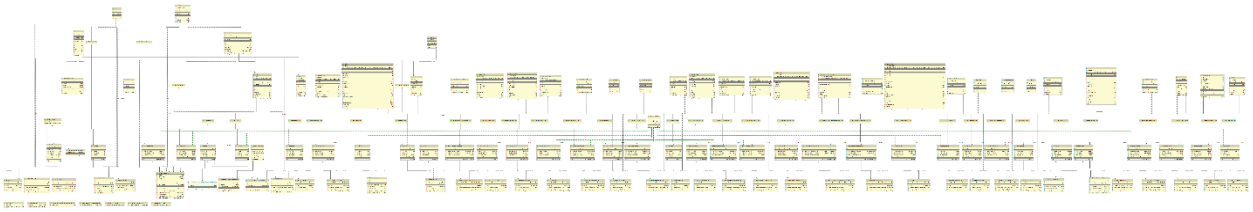
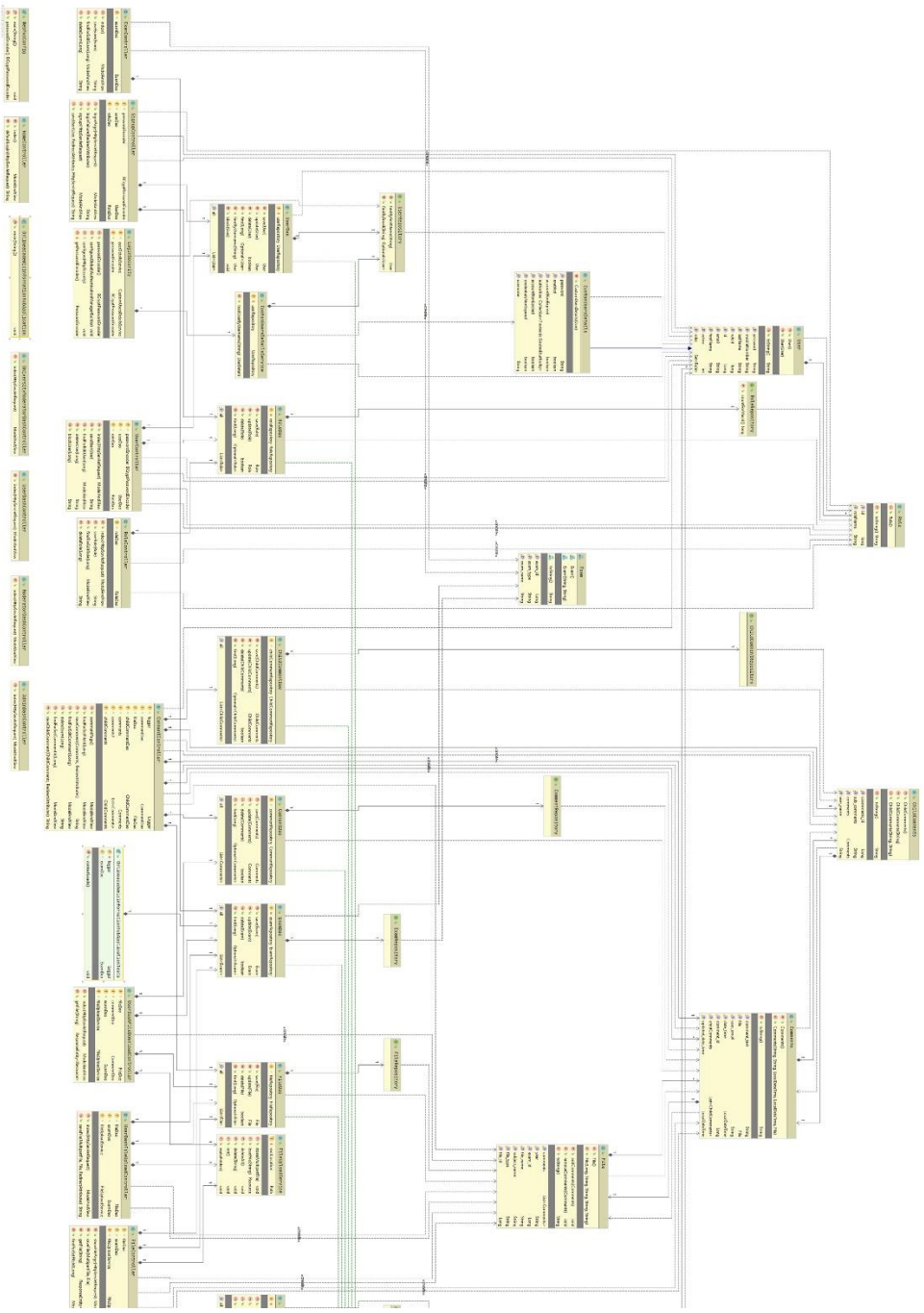


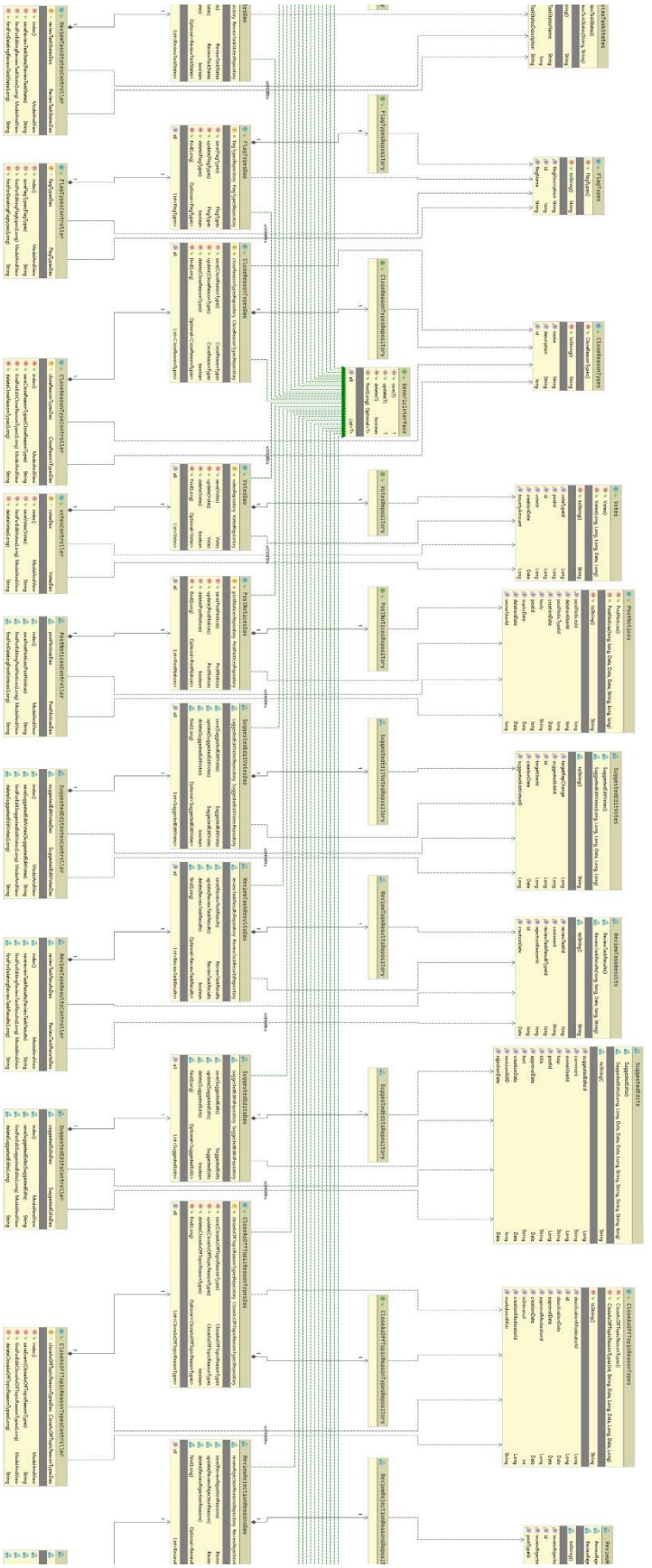
Figure 10:class diagram.

©Daffodil International University



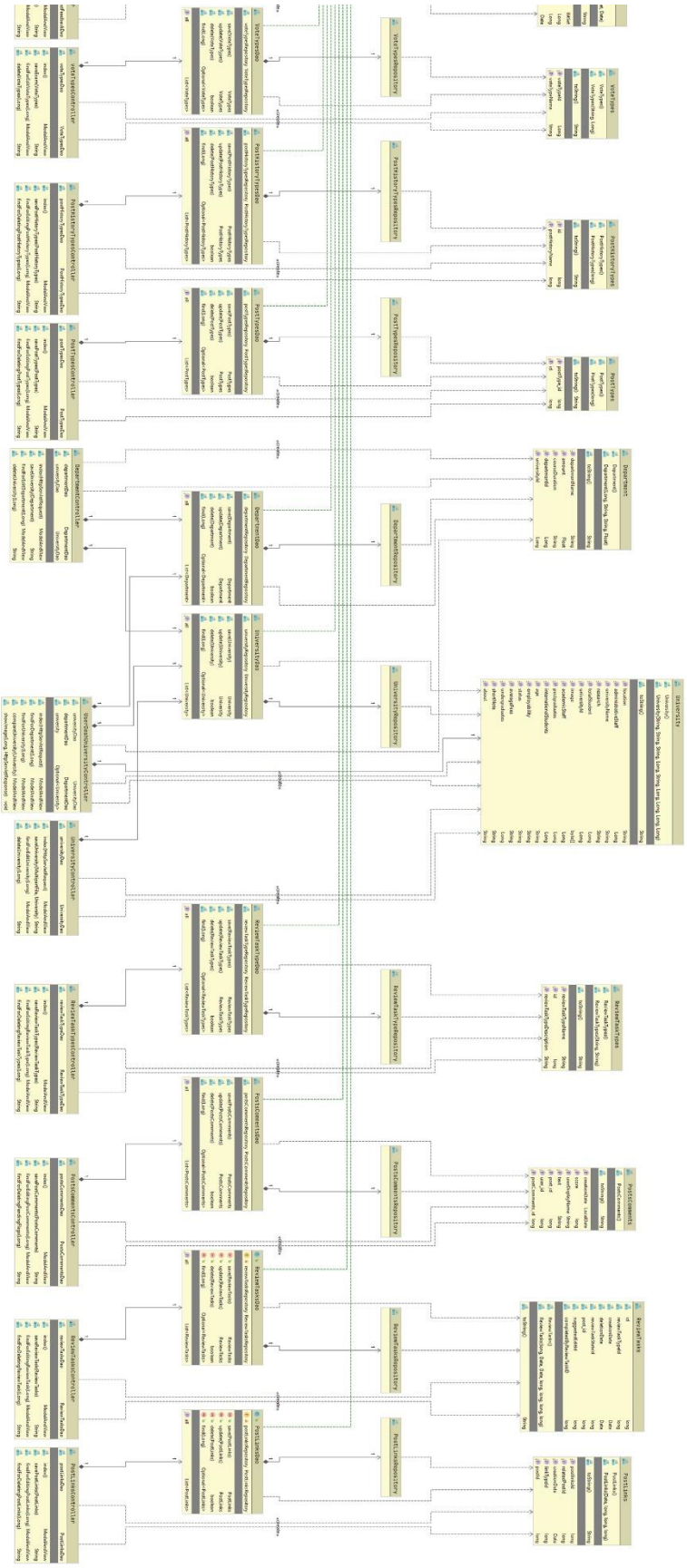
©Daffodil International University

©Daffodil International University



©Daffodil International University

©Daffodil International University



©Daffodil International University

Description:

Class diagram is that which contains variables, methods, classes, functions, working structure and shows the relationship between each one of them. It has a set of classes, interfaces, collaborations and their relationships. It shows the interactions between the classes that is used in this system. It represents the whole system in a diagram. This class diagram contains many classes like, university, department, user, user detail, files, exam, role etc. and all these class has so many attributes, methods.

Such as, university has university id, university name, location, total student, academic staff etc. Which shows the all information about a university. Department has, department id, department name, course duration, amount and also university id which act as a foreign key here. Department will represent the information of each university's information. User has user id, name, password, email, roles, role id etc. Role has role id, role name. Exam has exam type, exam name. All these classes with their attributes have relationships between them. By using this UML class diagram, we can show the whole system relationship how each of the table interact with each other.

User management system has the accessibility and security layer for every type of user. Admin, moderator, university moderator, user, pro user. All these users have different part of accessibility in the system. Admin has control over the whole system, moderator has access to the part which university and institution is going to be registered in the web application and show their details, they will manage all types of previous questions and answers of all education boards, and all university module information. There will be individual university moderator who will be managing specific university 's information which will be modified based on priority time and demand.

©Daffodil International University

Users can see discussions on the forum, question bank on the forum, talk with people, rate discussion topics, give answers, post job articles, post job exams, and many more. Users who are going to take part in these things will have to register without only seeing things posted in the forum. Everyone can post article in the forum and talk about. Moderators will be monitoring 24 hrs which post will going to be allowed or not in the forum, if there any bad comments are given or not, later which will be put into the AI to monitor each and every second if any bad comments are coming or not, which comments should be given priority or permitted to post. There will be a pro feature item in this web application. In this feature there will be thousand types of model tests to be given for getting prepared for the desired destination. Various types of exams, time duration, exam time schedule, school-college-university admission tests and many more can be given online. Various types of respective teachers from different institutions will be moderating the answer sheets. Pro features will be accessible after paying a short amount of cost.

©Daffodil International University

8.5 ERD Diagram

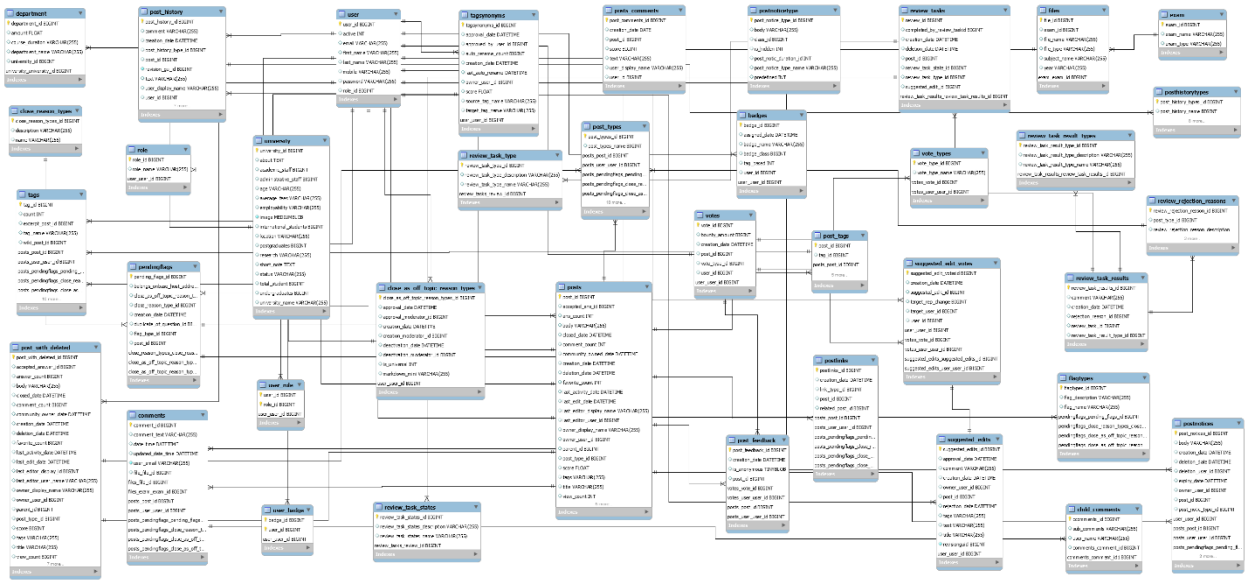


Figure 11:ERD Diagram.

Description:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.ER diagrams are used to sketch out the design of a database. Entity user has user_id BIGINT(20) primary key. Child_comments has comments id BIGINT(20) primary key.user role hase two primary key as user id and role id BIGINT(20).persistent logins has series primary key. university has university id. Department, user organization and finance all have their individual id as primary key. Role has role id as primary key. File has file id. Exam has exam id as primary id. Password reset token has also id has primarily. Organization entity has org id as

©Daffodil International University

primery key. Here entity User has many to 1 relation between entity user_role. Entity user has many to 1 relation between entity role. Entity user role has many to 1 relation to entity password_reset_token. Entity hibernate_sequence hase many to 1 relation between persistent_login. Child_comments entity has also many to 1 relation between entity comment.entity comment has 1 to many relation to entity files. Entity files has 1 to many relation to exams. Entity University has 1 to many relationship between entity department. Financial entity has 1 to many relation between organization.. entity organization has 1 to many relation with entity user_organization

Now here comes the forum part. There are different and a bunch of entities are utilized in this system. Badge id from badge entity is settled as foreign field in user section. Votes are going to take place through the posts of the forum. People can give an upvote or a downvote to individual post. One user one vote system has designed. Votes are also systemized with the type of vote people can give. The system will show suggestion about editing votes and a result of edited votes. People can give feedback about the post. Its different from comment section. There will be a trash systemization of closed thing topics, post, and more. The ‘Close as off topic reason types” will hold the details of users registered and blocked or deleted/banned with specific time calculation and the typical reason the user’s banned for. And a post with deleted section where all kinds of specific details of the post/post will be captured. How the post was, post type, comments and everything about it. Posts can be flagged or reposted by users based on some flag types or custom flagging types. When posts are flagged a review section as created for moderators and admins. Posts will be reviewed and a date will be stored for it. The flagged post will be reviewed by a systematic way which will follow some rules. After review a

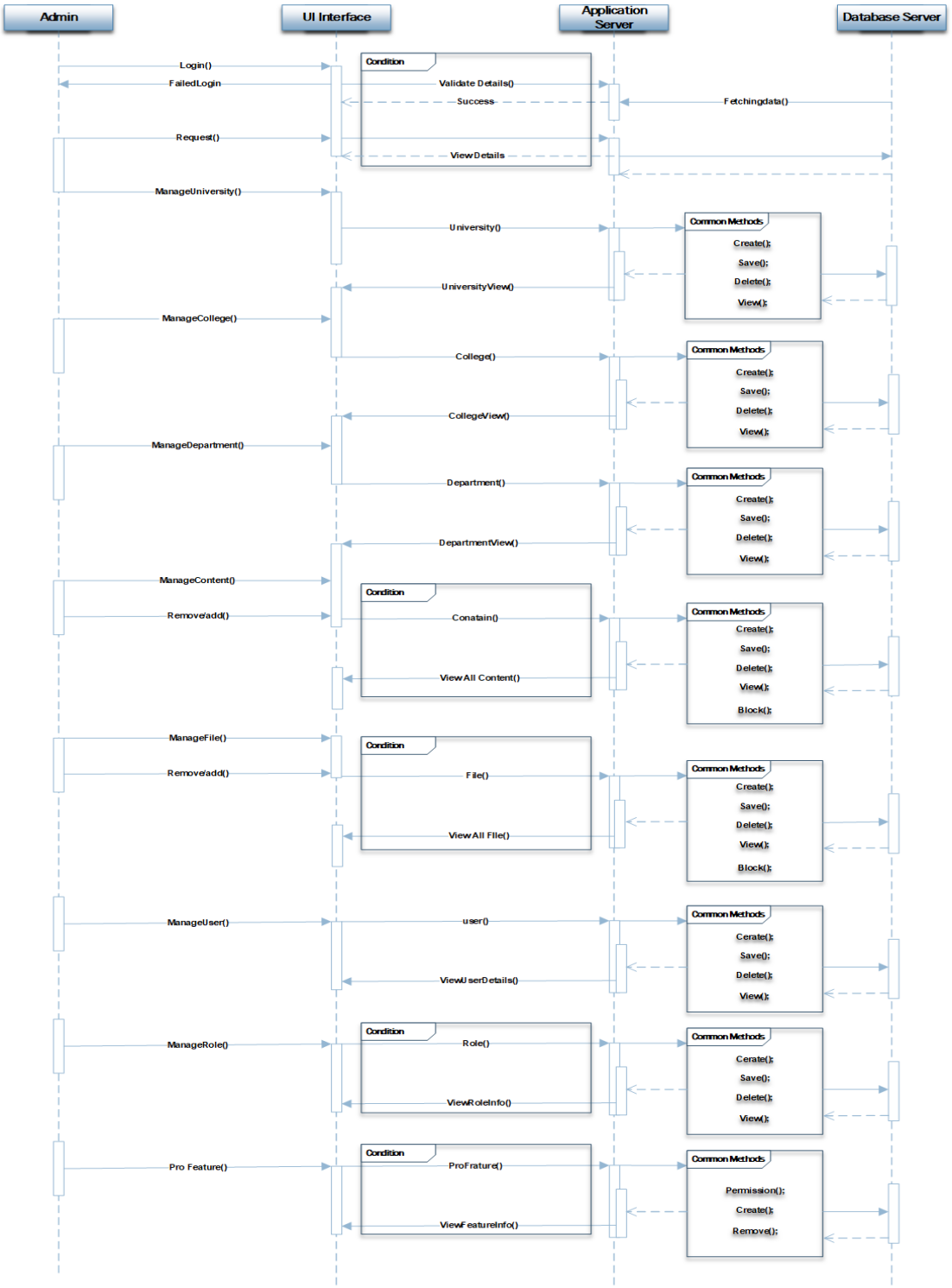
©Daffodil International University

detailed information about the review will be stored for future consultation. Furthermore, a Post History section will be stored and monetized. Every post will be categorized by post history types. Well now a very important part is Tagging posts with specific keywords for making it relevant with the discussion topics. Tags will be connected by post id. A synonym section of tags is also available for posts. Last but not the least the most important part is Post / Threads by which the Forum term is stablished. All the posts are utilized by user ids, post types, comments, post links, post notice, post notice types. All the Entities are very much connected to each other in the system.

©Daffodil International University

8.6 Sequence Diagram

Sequence diagram: As an admin



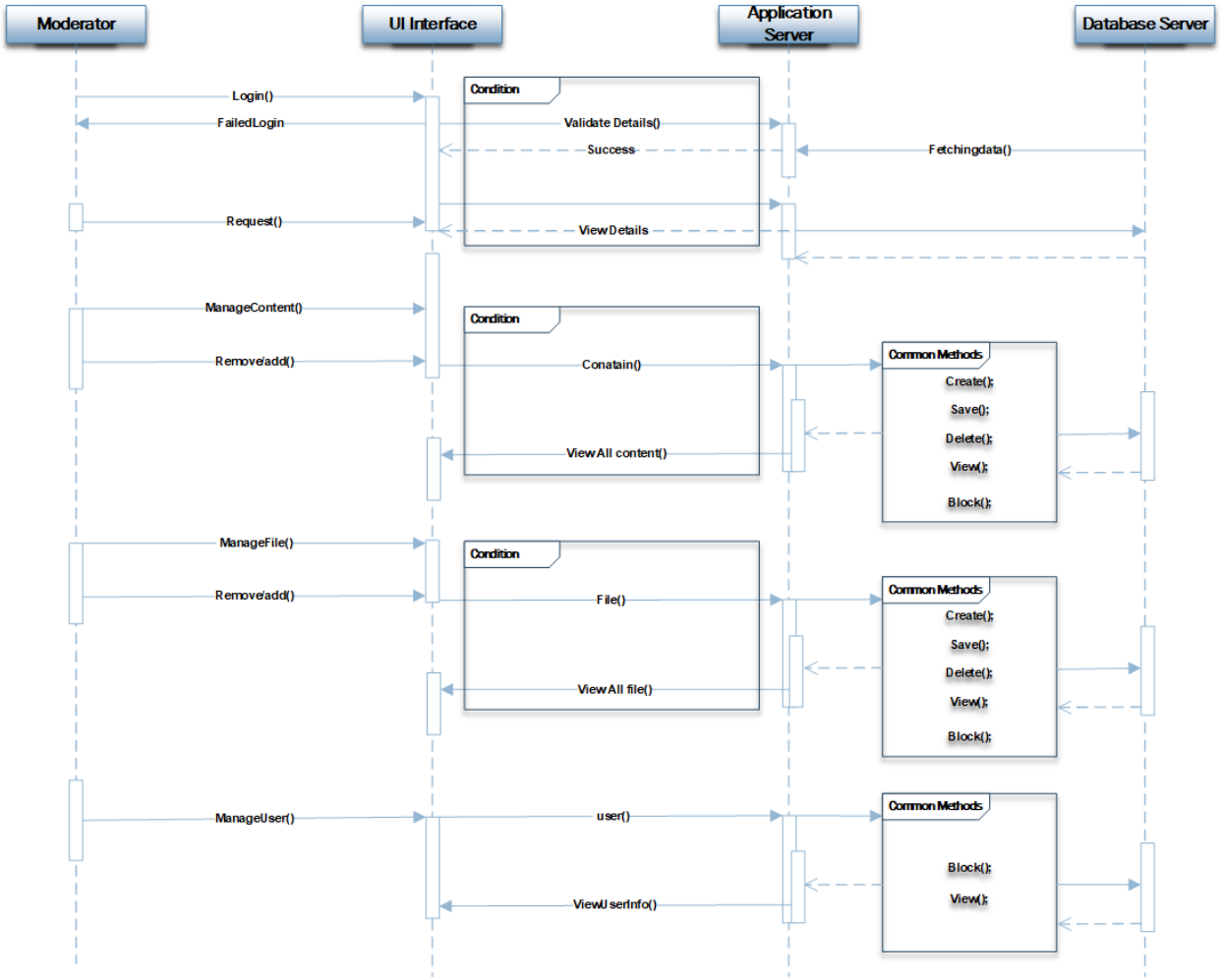
©Daffodil International University

Description:

This diagram shows that the admin request for login to the admin portal. It fetches the data which stored on the database server. Database stored the information of the school, college and university. Admin can view the detail information of the institution that are stored on the database server. Admin can manage the university, college, and department. And he can also remove any user from the system. The files which are uploaded by the user are managed by admin and he can manage or remove any files from the database server.

©Daffodil International University

As a moderator:

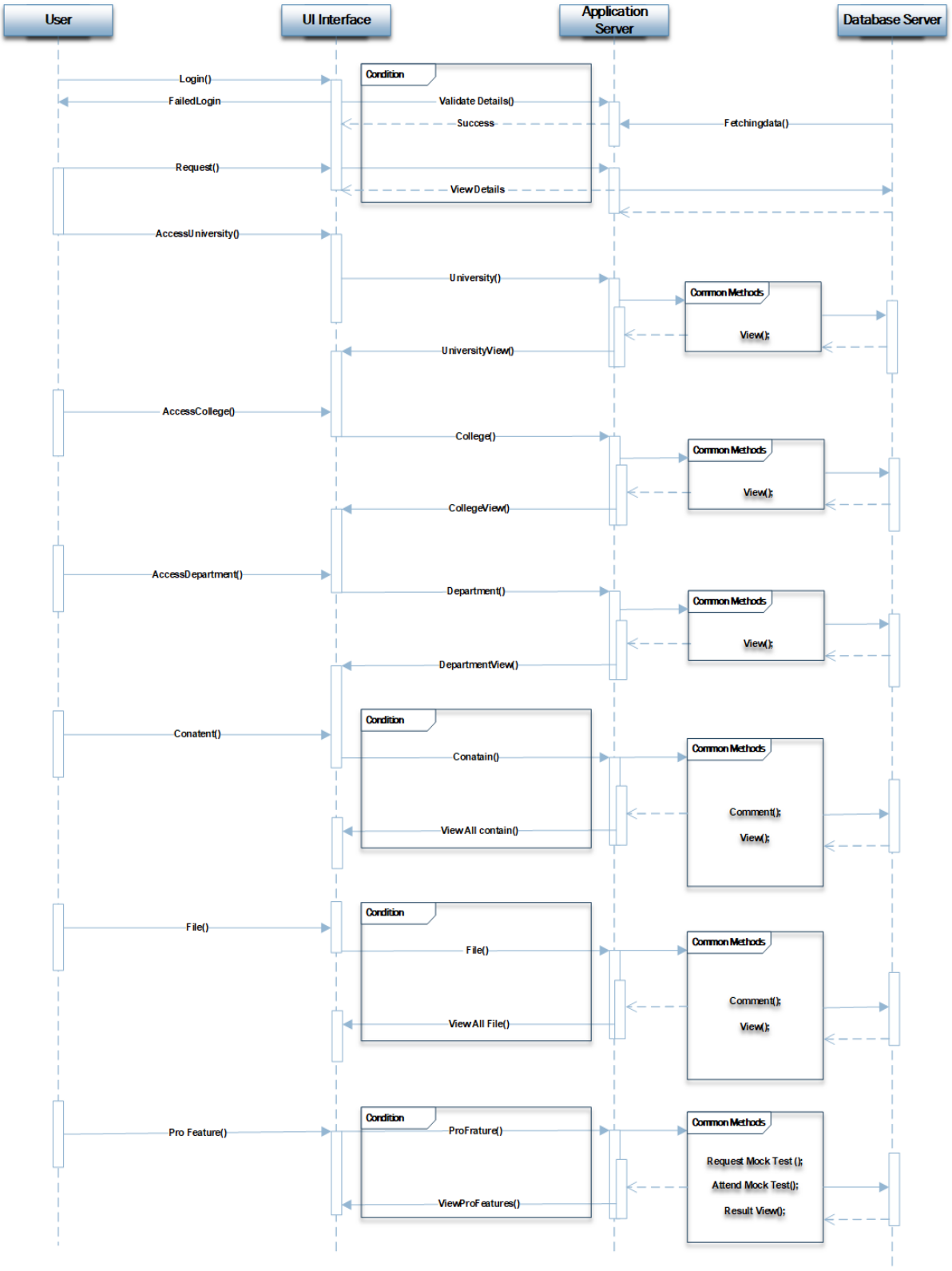


Description:

Moderator request to login into the system. It fetches data from database and give them login onto the system. Moderator can manage the files, remove any files from the server and he also manage the user. He can approve any user to use the system or block them from the system if any abusive is occurred by them.

©Daffodil International University

As a user:



©Daffodil International University

Description:

To login into the system as a user first they have to sign up. After that, the user information will be stored on the database. Every time when user request to login it fetch the data from the database server and give them approval to login. User can view the detail information of university, school or college. They can upload any file which are approved by the moderator. User can request for any test to participate on that. All these information, question bank and answer are store on database server. After any request it fetch the data from database and show the detail information to user.

©Daffodil International University

Chapter 9– Deployment / Development

9.1 Core Module Coding Samples

File Management

- **File Entity is Created**

```
@Entity
@Table(name = "files")
public class File {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "file_id")
    private Long file_id;

    @Column(name = "exam_id")
    private Long exam_id;

    @Column(name = "file_type")
    private String file_type;

    @Column(name = "file_name")
    private String file_name;

    @Column(name = "subject_name")
    private String subject_name;

    @Column(name = "year")
    private String year;


    @OneToMany(mappedBy = "file")
    private List<Comments> comments = new ArrayList<>();

    public File() {
    }

    public File(Long exam_id, String file_type, String file_name, String
subject_name, @NotEmpty(message = "*Please provide year") String year) {
        this.exam_id = exam_id;
        this.file_type = file_type;
        this.file_name = file_name;
        this.subject_name = subject_name;
        this.year = year;
    }
```


©Daffodil International University

- **Getter Setters for taking Data & Saving them into Database**

```
public Long getFile_id() {
    return file_id;
}

public void setFile_id(Long file_id) {
    this.file_id = file_id;
}

public Long getExam_id() {
    return exam_id;
}

public void setExam_id(Long exam_id) {
    this.exam_id = exam_id;
}

public String getFile_type() {
    return file_type;
}

public void setFile_type(String file_type) {
    this.file_type = file_type;
}

public String getFile_name() {
    return file_name;
}

public void setFile_name(String file_name) {
    this.file_name = file_name;
}

public String getSubject_name() {
    return subject_name;
}

public void setSubject_name(String subject_name) {
    this.subject_name = subject_name;
}

public String getYear() {
    return year;
}

public void setYear(String year) {
    this.year = year;
}

public List<Comments> getComments() {
    return comments;
}
```


©Daffodil International University

- **A Method is created to Show all those data**

```
    @Override
    public String toString() {
        return "File{" +
                "file_id=" + file_id +
                ", exam_id=" + exam_id +
                ", file_type='" + file_type + '\'' +
                ", file_name='" + file_name + '\'' +
                ", subject_name='" + subject_name + '\'' +
                '}';
    }
}
```

2.2 File Class

```
package ac.daffodil.repository;


import ac.daffodil.model.File;
import org.springframework.data.jpa.repository.JpaRepository;


public interface FileRepository extends JpaRepository<File, Long> {


}
```

2.3 File Dao

- **Here the Data Access Object class of file is controlling all the functionalities of file management**

```
@Repository
@Transactional
public class FileDao implements GenericInterface<File> {

    @Autowired
    FileRepository fileRepository;
```

- **Save file method**

```
    @Override
    public File save(File file) {
        fileRepository.save(file);
        return file;
    }
```


©Daffodil International University

- **Update file method**

```
    @Override
    public File update(File file) {
        fileRepository.save(file);
        return file;
    }
```

- **Delete file method**

```
    @Override
    public boolean delete(File file) {
        fileRepository.delete(file);
        return true;
    }
```

- **List out & show files method**

```
    @Override
    public List<File> getAll() {
        return fileRepository.findAll();
    }
```

- **Find specific file method**

```
    @Override
    public Optional<File> find(Long id) {
        return fileRepository.findById(id);
    }
}
```

2.4 userDashFileDownloadController Class

- **Here Is the code for controlling the file management / download for user**

```
@Controller
@RequestMapping("/user")
public class userDashFileDownloadController {

    @Autowired
    FileDao fileDao;

    @Autowired
    CommentDao commentDao;

    @Autowired
    ExamDao examDao;

    @Autowired
    FileUploadService fileUploadService;
```


©Daffodil International University

- **All file indexing method**

```
    @RequestMapping(value = {"/userDashFileDownloadPage"}, method =
RequestMethod.GET)
    public ModelAndView index(HttpServletRequest request) {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.addObject("newFile", new File());
        modelAndView.addObject("files", fileDao.getAll());
        modelAndView.addObject("exams", examDao.getAll());
        modelAndView.setViewName("user/userDashFileDownload");
        return modelAndView;
    }
```

- **File Fetching Method**

```
    @GetMapping("/files/{filename}")
    @ResponseBody
    public ResponseEntity<Resource> getFile(@PathVariable String filename) {
        Resource file = fileUploadService.loadFile(filename);
        return ResponseEntity.ok()
                .header(HttpHeaders.CONTENT_DISPOSITION, "attachment; filename=\"" +
file.getFilename() + "\"")
                .body(file);
    }
}
```

2.5 userDashFileUploadController Class

```
@Controller
@RequestMapping("/user")
public class userDashFileUploadController {

    @Autowired
    FileDao fileDao;

    @Autowired
    ExamDao examDao;

    @Autowired
    FileUploadService fileUploadService;
```


©Daffodil International University

- **File representing method**

```
    @RequestMapping(value = {"/userDashFileUploadPage"}, method = RequestMethod.GET)
    public ModelAndView index(HttpServletRequest request) {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.addObject("newFile", new File());
        modelAndView.addObject("exams", examDao.getAll());
        modelAndView.setViewName("user/userDashFileUpload");
        return modelAndView;
    }
```

- **Saving file method in hard disk maintaining sequence**

```
    @RequestMapping(value = "/saveFile", method = RequestMethod.POST)
    public String saveFile(@RequestParam("file") MultipartFile file, File newFile,
RedirectAttributes redirectAttributes) {

        try {
            fileUploadService.store(file);
            newFile.setFile_name(file.getOriginalFilename());
            fileDao.save(newFile);
            redirectAttributes.addFlashAttribute("message", "Your File Upload
Successfully...");
            redirectAttributes.addFlashAttribute("alertClass", "alert-success");
        } catch (Exception e) {
            redirectAttributes.addFlashAttribute("message", e.getMessage());
            redirectAttributes.addFlashAttribute("alertClass", "alert-danger");
        }

        return "redirect:/user/userDashFileUploadPage";

    }
}
```

2.5 File Upload Service Class

Here the files how they will be uploaded the service code is written

©Daffodil International University

```
@Service
public class FileUploadService {

    private final Path rootLocation = Paths.get("upload-dir");

    public void store(MultipartFile file) {
        try {
            Files.copy(file.getInputStream(),
this.rootLocation.resolve(file.getOriginalFilename()));
        } catch (Exception e) {
            throw new RuntimeException("FAIL!");
        }
    }

    public Resource loadFile(String filename) {
        try {
            Path file = rootLocation.resolve(filename);
            Resource resource = new UrlResource(file.toUri());
            if (resource.exists() || resource.isReadable()) {
                return resource;
            } else {
                throw new RuntimeException("FAIL!");
            }
        } catch (MalformedURLException e) {
            throw new RuntimeException("FAIL!");
        }
    }

    public void deleteAll() {
        FileSystemUtils.deleteRecursively(rootLocation.toFile());
    }

    public void delete(String filename) throws IOException {
        FileSystemUtils.deleteRecursively(rootLocation.resolve(filename).toFile());
    }

    public void init() {
```


©Daffodil International University

2.6 FileController Class

- **Here the file controller methods are written**

```
@Controller
@RequestMapping("/file")
public class FileController {

    @Autowired
    FileDao fileDao;

    @Autowired
    ExamDao examDao;

    @Autowired
    FileUploadService fileUploadService;
```

- **Show file page & it's options for file management by fetching them from database**

```
    @RequestMapping(value = "/filePage", method = RequestMethod.GET)
    public ModelAndView showFilePage(HttpServletRequest request) {
        fileUploadService.makeFolder();
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.addObject("message", request.getParameter("message"));
        modelAndView.addObject("newFile", new File());
        modelAndView.addObject("exams", examDao.getAll());
        modelAndView.addObject("files", fileDao.getAll());
        modelAndView.setViewName("admin/adminFile");
        return modelAndView;
    }
```

- **Save file method**

©Daffodil International University

```
@RequestMapping(value = "/saveFile", method = RequestMethod.POST)
public String saveFile(@RequestParam("file") MultipartFile file, File newFile) {
    ModelAndView modelAndView = new ModelAndView();
    try {
        fileUploadService.store(file);
        newFile.setFile_name(file.getOriginalFilename());
        fileDao.save(newFile);
        modelAndView.addObject("message", "Data Has Been Saved...");
    } catch (Exception e) {
        modelAndView.addObject("message", "Data Hasn`t Been Saved Properly...");
    }
    return "redirect:/file/filePage";
}
```

- **Get File from user**

```
    @GetMapping("/files/{filename}")
    @ResponseBody
    public ResponseEntity<Resource> getFile(@PathVariable String filename) {
        Resource file = fileUploadService.loadFile(filename);
        return ResponseEntity.ok()
                .header(HttpHeaders.CONTENT_DISPOSITION, "attachment;
filename=\"" + file.getFilename() + "\"")
                .body(file);
    }
```

3 Exam Management

3.1 Exam Class

- **This class is creating for creating an entity named exam and to save exam details in database by following some methods, objects, variables**

©Daffodil International University

```java
package ac.daffodil.model;

import javax.persistence.*;
import javax.validation.constraints.NotEmpty;


@Entity
@Table(name = "exam")
public class Exam {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "exam_id")
    private Long exam_id;


    @Column(name = "exam_type")
    @NotEmpty(message = "*please select an examtype")
    private String exam_type;

    @Column(name = "exam_name")
    @NotEmpty(message = "*please select exam name")
    private String exam_name;
```

- **method for selecting exam type**

```java
    public Exam() {
    }

    public Exam(@NotEmpty(message = "*please select an examtype") String exam_type,
@NotEmpty(message = "*please select exam name") String exam_name) {
        this.exam_type = exam_type;
        this.exam_name = exam_name;
    }
```

- **taking inputs for exam files**

©Daffodil International University

```
    public Long getExam_id() {
        return exam_id;
    }

    public void setExam_id(Long exam_id) {
        this.exam_id = exam_id;
    }

    public String getExam_type() {
        return exam_type;
    }

    public void setExam_type(String exam_type) {
        this.exam_type = exam_type;
    }

    public String getExam_name() {
        return exam_name;
    }

    public void setExam_name(String exam_name) {
        this.exam_name = exam_name;
    }
```

- **Showing exam details**

```
    @Override
    public String toString() {
        return "Exam{" +
                "exam_id=" + exam_id +
                ", exam_type='" + exam_type + '\'' +
                ", exam_name='" + exam_name + '\'' +
                '}';
    }
}
```

3.2 Exam Repository

- **Exam JPA dependency repository inheritance**

©Daffodil International University

```
package ac.daffodil.repository;


import ac.daffodil.model.Exam;
import org.springframework.data.jpa.repository.JpaRepository;


public interface ExamRepository extends JpaRepository<Exam, Long> {
}
```

3.3 Exam Dao

```
@Repository
@Transactional
public class ExamDao implements GenericInterface<Exam> {
    @Autowired
    private ExamRepository examRepository;
```

- **Save exam information method**

```
    @Override
    public Exam save(Exam exam) {
        examRepository.save(exam);
        return exam;
    }
```

- **Update exam information method**

```
@Override
public Exam update(Exam exam) {
    examRepository.save(exam);
    return exam;
}
```

- **Delete exam information method**

```
    @Override
    public boolean delete(Exam exam) {
        examRepository.delete(exam);
        return true;
    }
```

- **Fetch all exam information method**

```
    @Override
    public List<Exam> getAll() {
        return examRepository.findAll();
    }
```


©Daffodil International University

- **Search exam information method**

```
    @Override
    public Optional<Exam> find(Long id) {
        return examRepository.findById(id);
    }
}
```

3.4 Exam Controller

- **Imported examDao Class to access methods from it**

```
@Controller
public class ExamController {

    @Autowired
    ExamDao examDao;
```

- **Exam details representation method**

```
    @RequestMapping(value = {"/exam"}, method = RequestMethod.GET)
    public ModelAndView index() {
        ModelAndView modelAndView = new ModelAndView();
        Exam newExam = new Exam();
        modelAndView.addObject("newExam", newExam);
        modelAndView.addObject("exams", examDao.getAll());
        modelAndView.setViewName("admin/adminExam");
        return modelAndView;
    }
```

- **Save Exam details method**

```
    @RequestMapping(value = {"/exam/save"}, method = RequestMethod.POST)
    public String saveExam(Exam exam) {
        examDao.save(exam);
        return "redirect:/exam";
    }
```

- **Editing exam information method**

©Daffodil International University

```
    @RequestMapping(value = {"/exam/find/{exam_id}"}, method = RequestMethod.GET)
    public ModelAndView findForEditExam(@PathVariable(required = true, name =
"exam_id") Long exam_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<Exam> exam = examDao.find(exam_id);
        modelAndView.addObject("newExam", exam.get());
        modelAndView.addObject("exams", examDao.getAll());
        modelAndView.setViewName("admin/adminExam");
        return modelAndView;
    }
```

- **Deleting exam information method**

```
    @RequestMapping(value = "/exam/delete/{exam_id}", method = RequestMethod.GET)
    public String deleteExam(@PathVariable(required = true, name = "exam_id") Long
exam_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<Exam> exam = examDao.find(exam_id);
        examDao.delete(exam.get());
        modelAndView.addObject("message", " Data Has Been Deleted...");
        return "redirect:/exam";
    }
}
```

3.5 Comments

- **Comment Class is for taking comments upon files & Exam**

©Daffodil International University

```java
@Entity
@Table(name = "comments")
public class Comments {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long comment_id;


    @Column(nullable = false)
    private String user_email;

    @NotEmpty
    private String comment_text;

    @CreationTimestamp
    private LocalDateTime date_time;

    @UpdateTimestamp
    private LocalDateTime updated_date_time;

    @ManyToOne
    private File file;

    @OneToMany(mappedBy = "comments")
    private List<ChildComments> childComments = new ArrayList<>();
```

- **Taking comments as input**

©Daffodil International University

```
    public Comments() {
    }


    public Comments(String user_email, String comment_text, LocalDateTime date_time,
LocalDateTime updated_date_time, File file) {
        this.user_email = user_email;
        this.comment_text = comment_text;
        this.date_time = date_time;
        this.updated_date_time = updated_date_time;
        this.file = file;
    }


    public Long getComment_id() {
        return comment_id;
    }

    public void setComment_id(Long comment_id) {
        this.comment_id = comment_id;
    }

    public String getComment_text() {
        return comment_text;
    }

    public void setComment_text(String comment_text) {
        this.comment_text = comment_text;
    }

    public LocalDateTime getDate_time() {
        return date_time;
    }

    public void setDate_time(LocalDateTime date_time) {
        this.date_time = date_time;
    }

    public LocalDateTime getUpdated_date_time() {
        return updated_date_time;
    }

    public File getFile() {
        return file;
    }

    public void setFile(File file) {
        this.file = file;
    }

    public String getUser_email() {
```


©Daffodil International University

- **Showing Comments method**

```
    @Override
    public String toString() {
        return "Comments{" +
                "comment_id=" + comment_id +
                ", user_email='" + user_email + '\'' +
                ", comment_text='" + comment_text + '\'' +
                ", date_time=" + date_time +
                ", updated_date_time=" + updated_date_time +
                ", file=" + file +
                ", childComments=" + childComments +
                '}';
    }
}
```

3.6 Child Comments

- **Child Comment Class is for taking Child comments or comments of comments upon files & Exam**
- **Taking Child comments as input**

```
package ac.daffodil.model;
import javax.persistence.*;
@Entity
public class ChildComments {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long ccomments_id;
    private String sub_comments;
    private String user_name
    @ManyToOne
    private Comments comments;
    public ChildComments() {
    }
```


©Daffodil International University

```
    public ChildComments(String sub_comments) {
        this.sub_comments = sub_comments;
    }
    public ChildComments(String sub_comments, String user_name) {
        this.sub_comments = sub_comments;
        this.user_name = user_name;
    }
    public Long getCcomments_id() {
        return ccomments_id;
    }
    public void setCcomments_id(Long ccomments_id) {
        this.ccomments_id = ccomments_id;
    }

    public String getSub_comments() {
        return sub_comments;
    }
    public void setSub_comments(String sub_comments) {
        this.sub_comments = sub_comments;
    }
    public Comments getComments() {
        return comments;
    }
    public void setComments(Comments comments) {
        this.comments = comments;
    }
    public String getUser_name() {
        return user_name;
    }
    public void setUser_name(String user_name) {
        this.user_name = user_name;
    }
```

Showing child comments method

©Daffodil International University

```
    @Override
    public String toString() {
        return "ChildComments{" +
                "ccomments_id=" + ccomments_id +
                ", sub_comments='" + sub_comments + '\'' +
                ", user_name='" + user_name + '\'' +
                ", comments=" + comments +
                '}';
    }
}
```

3.7 Comment Repository

- **SpringBoot JPA repository**

```
package ac.daffodil.repository;
import ac.daffodil.model.Comments;
import org.springframework.data.jpa.repository.JpaRepository;
public interface CommentRepository extends JpaRepository<Comments, Long> {
}
```

3.8 Child Comments Repository

- **SpringBoot JPA repository**

```
package ac.daffodil.repository;
import ac.daffodil.model.ChildComments;
import org.springframework.data.jpa.repository.JpaRepository;
public interface ChildCommentRepository extends JpaRepository<ChildComments, Long> {
}
```

3.9 Comments Dao

```
@Repository
@Transactional
public class CommentDao implements GenericInterface<Comments> {
    @Autowired
    CommentRepository commentRepository;
```

- **Saving Comments into database method**

```
    @Override
    public Comments save(Comments comments) {
        commentRepository.save(comments);
        return comments;
    }
```


©Daffodil International University

- **updating Comments into database method**

```
    @Override
    public Comments update(Comments comments) {
        commentRepository.save(comments);
        return comments;
    }
```

- **Deleting Comments from database method**

```
    @Override
    public boolean delete(Comments comments) {
        commentRepository.delete(comments);
        return true;
    }
```

- **Showing Comments from database method**

```
    @Override
    public List<Comments> getAll() {
        return commentRepository.findAll();
    }
```

- **Finding Comments from database method**

```
    @Override
    public Optional<Comments> find(Long id) {
        return commentRepository.findById(id);
    }
}
```

3.10 Child Comments Dao

```
@Repository
public class ChildCommentDao implements GenericInterface<ChildComments> {

    @Autowired
    ChildCommentRepository childCommentRepository;
```

- **Saving Child Comments into database method**

©Daffodil International University

```
    @Override
    public ChildComments save(ChildComments childComments) {
        childCommentRepository.save(childComments);
        return childComments;
    }
```

- **Updating Child Comments into database method**

```
    @Override
    public ChildComments update(ChildComments childComments) {
        childCommentRepository.save(childComments);
        return childComments;
    }
```

- **Deleting Child Comments from database method**

```
    @Override
    public boolean delete(ChildComments childComments) {
        childCommentRepository.delete(childComments);
        return true;
    }
```

- **Showing Child Comments from database method**

```
    @Override
    public List<ChildComments> getAll() {
        return childCommentRepository.findAll();
    }
```

- **Finding Child Comments from database method**

```
@Override
    public Optional<ChildComments> find(Long id) {
        return childCommentRepository.findById(id);
    }
}
```


©Daffodil International University

3.11 Comment Controller

```
@Controller
public class CommentController {

    Logger logger = LoggerFactory.getLogger(getClass());

    @Autowired
    CommentDao commentDao;

    @Autowired
    FileDao fileDao;

    @Autowired
    ChildCommentDao childCommentDao;

    Comments comments = new Comments();
    List<Comments> comments1 = new LinkedList<>();

    ChildComments childComments = new ChildComments();
```

- **Comment Representation method**

```
@RequestMapping(value = {"/comment"}, method = RequestMethod.GET)
public ModelAndView commentPage() {
    ModelAndView modelAndView = new ModelAndView();
    Comments newComment = new Comments();

    comments1 = new LinkedList<>();
    for (Comments cmt : commentDao.getAll()) {
        if (cmt.getFile().getFile_id() == comments.getFile().getFile_id()) {
            comments1.add(cmt);
        }
    }

    modelAndView.addObject("newComment", comments);
    modelAndView.addObject("commentList", comments1);
    modelAndView.setViewName("user/userDashComment");
    return modelAndView;
}
```

- **Find & Get Comment & File ID for resolving problems**

©Daffodil International University

```
    @RequestMapping(value = {"/comment/findForFile/{file_id}"}, method =
RequestMethod.GET)
    public ModelAndView findForSetFileId(@PathVariable(required = true, name =
"file_id") Long file_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<File> file = fileDao.find(file_id);
        comments.setFile(file.get());

        Object principal =
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
        if (principal instanceof User) {
            String email = ((User) principal).getEmail();
            comments.setUser_email(email);
        }

        comments1 = new LinkedList<>();
        for (Comments cmt : commentDao.getAll()) {
            if (cmt.getFile().getFile_id() == file_id) {
                comments1.add(cmt);
            }
        }

        modelAndView.addObject("commentList", comments1);
        modelAndView.addObject("newComment", comments);
        modelAndView.setViewName("user/userDashComment");
        return modelAndView;
    }
```

Saving Comments into database method

```
    @RequestMapping(value = {"/comment/saveComment"}, method = RequestMethod.POST)
    public String saveComment(Comments comments, RedirectAttributes
redirectAttributes) {
        commentDao.save(comments);
        redirectAttributes.addFlashAttribute("message", "You Comment is= " +
comments.getComment_text());
        redirectAttributes.addFlashAttribute("alertClass", "alert-success");
        return "redirect:/comment";
    }
```


©Daffodil International University

- **Child Comment through Id**

```
    @RequestMapping(value = {"/comment/find/{comment_id}"}, method =
RequestMethod.GET)
    public ModelAndView findForEditComment(@PathVariable(required = true, name =
"comment_id") Long comment_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<Comments> comments = commentDao.find(comment_id);
        modelAndView.addObject("newComment", comments.get());
        modelAndView.addObject("commentList", comments1);
        modelAndView.setViewName("user/userDashComment");
        return modelAndView;
    }
```

- **Deleting Child Comments from database method**

```
 @RequestMapping(value = "/comment/delete/{comment_id}", method =
RequestMethod.GET)
    public String deleteExam(@PathVariable(required = true, name =
"comment_id") Long comment_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<Comments> comments = commentDao.find(comment_id);
        commentDao.delete(comments.get());
        modelAndView.addObject("message", " Data Has Been Deleted...");
        return "redirect:/comment";
    }
```

- **Child Comment through sequential id by finding comment id**

```
    @RequestMapping(value = {"/findForComment/{comment_id}"}, method =
RequestMethod.GET)
    public ModelAndView findForSetCommentId(@PathVariable(required = true, name =
"comment_id") Long comment_id) {
        ModelAndView modelAndView = new ModelAndView();
        Optional<Comments> comment = commentDao.find(comment_id);
        ChildComments childComments = new ChildComments();
        childComments.setComments(comment.get());
        modelAndView.addObject("newComment", childComments);
        System.out.println(childComments.getComments().getComment_id());
        modelAndView.setViewName("user/childComment");
        return modelAndView;
    }
```


©Daffodil International University

Saving Child Comment

```
    @RequestMapping(value = {"/comment/saveChildComment"}, method =
RequestMethod.POST)
    public String saveChildComment(ChildComments childComments,
RedirectAttributes redirectAttributes) {
        Object principal =
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
        if (principal instanceof User) {
            String name = ((User) principal).getFirstName();
            childComments.setUser_name(name);
            childCommentDao.save(childComments);
            redirectAttributes.addFlashAttribute("message", "Your Comment is=
" + comments.getComment_text());
            redirectAttributes.addFlashAttribute("alertClass", "alert-
success");
            return "redirect:/comment";
        }
        return "redirect:/comment";
    }
}
```

Forum management

Voting code samples

1.1 Votes Controller

1.1.1 Request mapper & Controller assigned

```
@Controller
@RequestMapping("/vote")
public class votesController {
  @Autowired
  VotesDao votesDao;
```

1.1.2 All votes will be viewed here

©Daffodil International University

```
  @RequestMapping(value = { "/votes" }, method = RequestMethod.GET)
  public ModelAndView index() {
    ModelAndView modelAndView = new ModelAndView();
    Votes votes = new Votes();
    modelAndView.addObject("newVotes", votes);
    modelAndView.addObject("votes", votesDao.getAll());
    modelAndView.setViewName("admin/adminVotes");
    return modelAndView;
  }
```

1.1.3 Saving a vote of a user

```
  @RequestMapping(value = { "/votes/save" }, method = RequestMethod.POST)
  public String saveVotes(Votes votes) {
    votesDao.save(votes);
    return "redirect:/votes";
  }
```

1.1.4 Finding a vote of a user (will not be required in later development)

```
  @RequestMapping(value={"/votes/find/{VoteId}"}, method = RequestMethod.GET)
  public ModelAndView findForEditVotes(@PathVariable(required = true, name = "VoteId") Long VoteId) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<Votes> votes= votesDao.find(VoteId);
    modelAndView.addObject("newVotes", votes.get());
    modelAndView.addObject("votes", votesDao.getAll());
    modelAndView.setViewName("admin/adminVotes");
    return modelAndView;
  }
```

1.1.5 Removing vote for the post

```
  @RequestMapping(value="/votes/delete/{VoteId}", method = RequestMethod.GET)
  public String deleteVotes(@PathVariable(required = true, name = "VoteId") Long VoteId) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<Votes> votes= votesDao.find(VoteId);
    votesDao.delete(votes.get());
    modelAndView.addObject("message", " Data Has Been Deleted...");
    return "redirect:/votes";
  }
}
```

1.2 Vote Types Controller

1.2.1 Request mapper & Controller assigned

```
@Controller
@RequestMapping("/voteType")
public class voteTypesController {
  @Autowired
  VoteTypesDao voteTypesDao;
```

1.2.2 Type of the votes are viewed

©Daffodil International University

```
  @RequestMapping(value = { "/voteTypes" }, method = RequestMethod.GET)
  public ModelAndView index() {
    ModelAndView modelAndView = new ModelAndView();
    VoteTypes voteTypes = new VoteTypes();
    modelAndView.addObject("newVoteTypes", voteTypes);
    modelAndView.addObject("voteTypes", voteTypesDao.getAll());
    modelAndView.setViewName("admin/adminVoteTypes");
    return modelAndView;
  }
```

1.2.3 A new vote type is saved

```
  @RequestMapping(value = { "/voteTypes/save" }, method = RequestMethod.POST)
  public String saveExam(VoteTypes voteTypes) {
     voteTypesDao.save(voteTypes);
     return "redirect:/voteTypes";
  }
```

1.2.4 Finding a saved vote type

```
  @RequestMapping(value={"/voteTypes/find/{VoteTypeId}"}, method = RequestMethod.GET)
  public ModelAndView findForEditVoteTypes(@PathVariable(required = true, name = "VoteTypeId") Long
VoteTypeId) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<VoteTypes> voteTypes= voteTypesDao.find(VoteTypeId);
    modelAndView.addObject("newVoteTypes", voteTypes.get());
    modelAndView.addObject("voteTypes", voteTypesDao.getAll());
    modelAndView.setViewName("admin/adminVoteTypes");
    return modelAndView;
  }
```

1.2.5 Deleting a vote type

```
  @RequestMapping(value="/voteTypes/delete/{VoteTypeId}", method = RequestMethod.GET)
  public String deleteVoteTypes(@PathVariable(required = true, name = "VoteTypeId") Long VoteTypeId) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<VoteTypes> voteTypes= voteTypesDao.find(VoteTypeId);
    voteTypesDao.delete(voteTypes.get());
    modelAndView.addObject("message", " Data Has Been Deleted...");
    return "redirect:/voteTypes";
  }
}
```

1.3 Suggested Edit Votes

1.3.1 Controller & Request mapper assigned

```
@Controller
@RequestMapping("/suggestedEditVote")
public class SuggestedEditVotesController {
  @Autowired
  SuggestedEditVotesDao suggestedEditVotesDao;
```

1.3.2 Viewing suggestion for votes

©Daffodil International University

```
  @RequestMapping(value = { "/suggestedEditVotes" }, method = RequestMethod.GET)
  public ModelAndView index() {
    ModelAndView modelAndView = new ModelAndView();
    SuggestedEditVotes suggestedEditVotes = new SuggestedEditVotes();
    modelAndView.addObject("newSuggestedEditVotes", suggestedEditVotes);
    modelAndView.addObject("suggestedEditVotes", suggestedEditVotesDao.getAll());
    modelAndView.setViewName("admin/adminSuggestedEditVotes");
    modelAndView.setViewName("user/userSuggestedEditVotes");
    return modelAndView;
  }
```

1.3.3 Saving a vote

```
  @RequestMapping(value = { "/suggestedEditVotes/save" }, method = RequestMethod.POST)
  public String saveSuggestedEditVotes(SuggestedEditVotes suggestedEditVotes) {
    suggestedEditVotesDao.save(suggestedEditVotes);
    return "redirect:/suggestedEditVotes";
  }
```

1.3.4 Editing a vote

```
  @RequestMapping(value={"/suggestedEditVotes/find/{SuggestedEditVotesID}"}, method =
RequestMethod.GET)
  public ModelAndView findForEditSuggestedEditVotes(@PathVariable(required = true, name =
"SuggestedEditVotesID") Long SuggestedEditVotesID) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<SuggestedEditVotes> suggestedEditVotes=
suggestedEditVotesDao.find(SuggestedEditVotesID);
    modelAndView.addObject("newSuggestedEditVotes", suggestedEditVotes.get());
    modelAndView.addObject("suggestedEditVotes", suggestedEditVotesDao.getAll());
    modelAndView.setViewName("admin/adminSuggestedEditVotes");
    modelAndView.setViewName("user/userSuggestedEditVotes");
    return modelAndView;
  }
```

1.3.5 Deleting a vote

```
  @RequestMapping(value="/suggestedEditVotes/delete/{SuggestedEditVotesID}", method =
RequestMethod.GET)
  public String deleteSuggestedEditVotes(@PathVariable(required = true, name = "SuggestedEditVotesID")
Long SuggestedEditVotesID) {
    ModelAndView modelAndView = new ModelAndView();
    Optional<SuggestedEditVotes> suggestedEditVotes=
suggestedEditVotesDao.find(SuggestedEditVotesID);
    suggestedEditVotesDao.delete(suggestedEditVotes.get());
    modelAndView.addObject("message", " Data Has Been Deleted...");
    return "redirect:/suggestedEditVotes";
  }
}
```

1.4 Suggested edit votes Dao

©Daffodil International University

1.4.1 Repository & model class imported

```
@Repository
@Transactional
public class SuggestedEditVotesDao implements GenericInterface<SuggestedEditVotes> {
```

1.4.2 All necessary methods from repository is used

```
  @Autowired
  private SuggestedEditVotesRepository suggestedEditVotesRepository;
  @Override
  public SuggestedEditVotes save(SuggestedEditVotes suggestedEditVotes) {
    suggestedEditVotesRepository.save(suggestedEditVotes);
    return suggestedEditVotes;
  }

  @Override
  public SuggestedEditVotes update(SuggestedEditVotes suggestedEditVotes) {
    suggestedEditVotesRepository.save(suggestedEditVotes);
    return suggestedEditVotes;
  }

  @Override
  public boolean delete(SuggestedEditVotes suggestedEditVotes) {
    suggestedEditVotesRepository.delete(suggestedEditVotes);
    return true;
  }

  @Override
  public List<SuggestedEditVotes> getAll() {
    return suggestedEditVotesRepository.findAll();
  }

  @Override
  public Optional<SuggestedEditVotes> find(Long id) {
    return suggestedEditVotesRepository.findById(id);
  }
}
```

1.5 Votes Dao

1.5.1 Repository & Model class imported

```
@Repository
@Transactional
public class VotesDao implements GenericInterface<Votes> {
```


©Daffodil International University

1.5.2 All necessary methods from repository is used

```
  @Autowired
  private VotesRepository votesRepository;

  @Override
  public Votes save(Votes votes) {
    votesRepository.save(votes);
    return votes;
  }

  @Override
  public Votes update(Votes votes) {
    votesRepository.save(votes);
    return votes;
  }

  @Override
  public boolean delete(Votes votes) {
    votesRepository.delete(votes);
    return true;
  }

  @Override
  public List<Votes> getAll() {
    return votesRepository.findAll();
  }

  @Override
  public Optional<Votes> find(Long id) {
    return votesRepository.findById(id);
  }
}
```

1.6 Vote types Dao

1.6.1 Repository & Model class imported

```
@Repository
@Transactional
public class VoteTypesDao implements GenericInterface<VoteTypes> {
```


©Daffodil International University

1.6.2 All necessary methods from repository is used

```
  @Autowired
  private VoteTypesRepository voteTypesRepository;
  @Override
  public VoteTypes save(VoteTypes voteTypes) {
    voteTypesRepository.save(voteTypes);
    return voteTypes;
  }

  @Override
  public VoteTypes update(VoteTypes voteTypes) {
    voteTypesRepository.save(voteTypes);
    return voteTypes;
  }

  @Override
  public boolean delete(VoteTypes voteTypes) {
    voteTypesRepository.delete(voteTypes);
    return true;
  }

  @Override
  public List<VoteTypes> getAll() {
    return voteTypesRepository.findAll();
  }

  @Override
  public Optional<VoteTypes> find(Long id) {
    return voteTypesRepository.findById(id);
  }
}
```

1.7 Suggested edit model class

1.7.1 Entity created

```
@Entity
@Table(name = "SuggestedEditVotes")
```


©Daffodil International University

1.7.2 Necessary fields are taken

```
public class SuggestedEditVotes {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "SuggestedEditVotesID")
  private Long SuggestedEditVotesID;

  @Column(name = "SuggestedEditId")
  private Long SuggestedEditId;

  @Column(name = "UserId")
  private Long id;

  @Column(name = "CreationDate")
  private Date CreationDate;

  @Column(name = "TargetUserId")
  private Long TargetUserId;

  @Column(name = "TargetRepChange")
  private Long TargetRepChange;

  public SuggestedEditVotes() {
  }
```


©Daffodil International University

1.7.3 Constructors created

```
  public SuggestedEditVotes(Long suggestedEditVotesID, Long suggestedEditId, Long id, Date creationDate,
Long targetUserId, Long targetRepChange) {
    SuggestedEditVotesID = suggestedEditVotesID;
    SuggestedEditId = suggestedEditId;
    this.id = id;
    CreationDate = creationDate;
    TargetUserId = targetUserId;
    TargetRepChange = targetRepChange;
  }

  public Long getSuggestedEditVotesID() {
    return SuggestedEditVotesID;
  }

  public void setSuggestedEditVotesID(Long suggestedEditVotesID) {
    SuggestedEditVotesID = suggestedEditVotesID;
  }

  public Long getSuggestedEditId() {
    return SuggestedEditId;
  }

  public void setSuggestedEditId(Long suggestedEditId) {
    SuggestedEditId = suggestedEditId;
  }

  public Long getId() {
    return id;
  }

  public void setId(Long id) {
    this.id = id;
  }

  public Date getCreationDate() {
    return CreationDate;
  }

  public void setCreationDate(Date creationDate) {
    CreationDate = creationDate;
  }

  public Long getTargetUserId() {
    return TargetUserId;
  }

  public void setTargetUserId(Long targetUserId) {
    TargetUserId = targetUserId;
  }
```


©Daffodil International University

```
  public Long getTargetRepChange() {
    return TargetRepChange;
  }

  public void setTargetRepChange(Long targetRepChange) {
    TargetRepChange = targetRepChange;
  }
```

1.7.4 Method for viewing all fields

```
  @Override
  public String toString() {
    return "SuggestedEditVotes{" +
          "SuggestedEditVotesID=" + SuggestedEditVotesID +
          ", SuggestedEditId=" + SuggestedEditId +
          ", id=" + id +
          ", CreationDate=" + CreationDate +
          ", TargetUserId=" + TargetUserId +
          ", TargetRepChange=" + TargetRepChange +
          '}';
  }
}
```

1.8 Votes Model class

1.8.1 Entity created

```
@Entity
@Table(name ="Votes")
public class Votes {
```

1.8.2 Necessary fields are taken

```
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "VoteId")
  private Long VoteId;

  @Column(name = "PostId")
  private Long PostId;

  @Column(name = "VoteTypeId")
  private Long VoteTypeId;

  @Column(name = "UserId")
  private Long id;

  @Column(name = "CreationDate")
  private Date CreationDate;

  @Column(name = "BountyAmount")
  private Long BountyAmount;
```

1.8.3 Constructors created

©Daffodil International University

```
public Votes() {
}
public Votes(Long postId, Long voteTypeId, Long id, Date creationDate, Long bountyAmount) {
   PostId = postId;
   VoteTypeId = voteTypeId;
   this.id = id;
   CreationDate = creationDate;
   BountyAmount = bountyAmount;
}
public Long getVoteId() {
   return VoteId;
}
public Long getVoteTypeId() {
   return VoteTypeId;
}
public void setVoteTypeId(Long voteTypeId) {
   VoteTypeId = voteTypeId;
}
public void setVoteId(Long voteId) {
   VoteId = voteId;
}
public Long getPostId() {
   return PostId;
}
public void setPostId(Long postId) {
   PostId = postId;
}
public Long getId() {
   return id;
}
public void setId(Long id) {
   this.id = id;
}
public Date getCreationDate() {
   return CreationDate;
}
public void setCreationDate(Date creationDate) {
   CreationDate = creationDate;
}
public Long getBountyAmount() {
   return BountyAmount;
}
public void setBountyAmount(Long bountyAmount) {
   BountyAmount = bountyAmount;
}
```

1.8.4 Method for viewing all fields

©Daffodil International University

```
  @Override
  public String toString() {
    return "Votes{" +
          "VoteId=" + VoteId +
          ", PostId=" + PostId +
          ", VoteTypeId=" + VoteTypeId +
          ", id=" + id +
          ", CreationDate=" + CreationDate +
          ", BountyAmount=" + BountyAmount +
          '}';
  }
}
```

1.9 Vote type model class

1.9.1 Entity created

```
@Entity
@Table(name = "VoteTypes")
public class VoteTypes {
```

1.9.2 Necessary fields are taken

```
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "VoteTypeId")
  private Long VoteTypeId;

  @Column(name = "VoteTypeName")
  private String VoteTypeName;
```

1.9.3 Constructors created

```
  public VoteTypes() {
  }
  public VoteTypes(String voteTypeName,Long voteTypeId) {
    VoteTypeName = voteTypeName;
    VoteTypeId = voteTypeId;
  }
  public Long getVoteTypeId() {
    return VoteTypeId;
  }
  public void setVoteTypeId(Long voteTypeId) {
    VoteTypeId = voteTypeId;
  }
  public String getVoteTypeName() {
    return VoteTypeName;
  }
  public void setVoteTypeName(String voteTypeName) {
    VoteTypeName = voteTypeName;
  }
```


©Daffodil International University

1.9.4 Method for viewing all fields

```
  @Override
  public String toString() {
    return "VoteTypes{" +
          "VoteTypeId=" + VoteTypeId +
          ", VoteTypeName='" + VoteTypeName + '\'' +
          '}';
  }
}
```

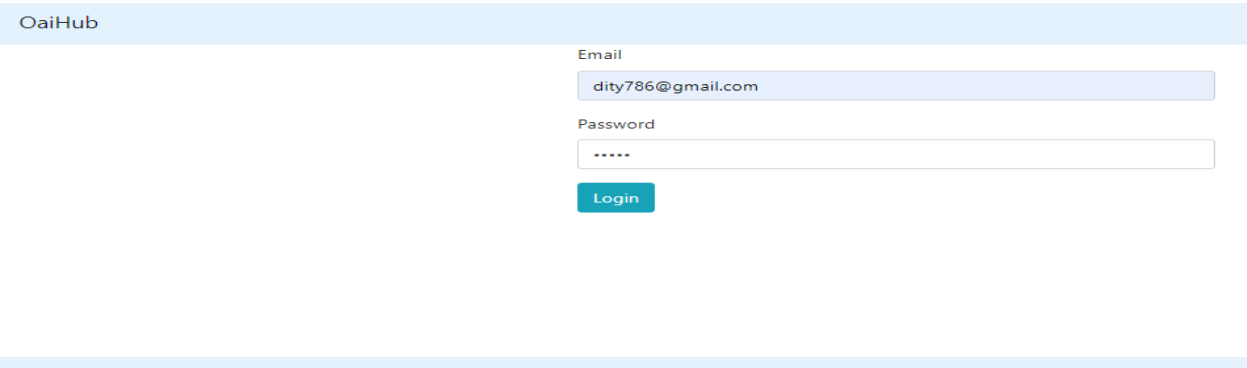

©Daffodil International University

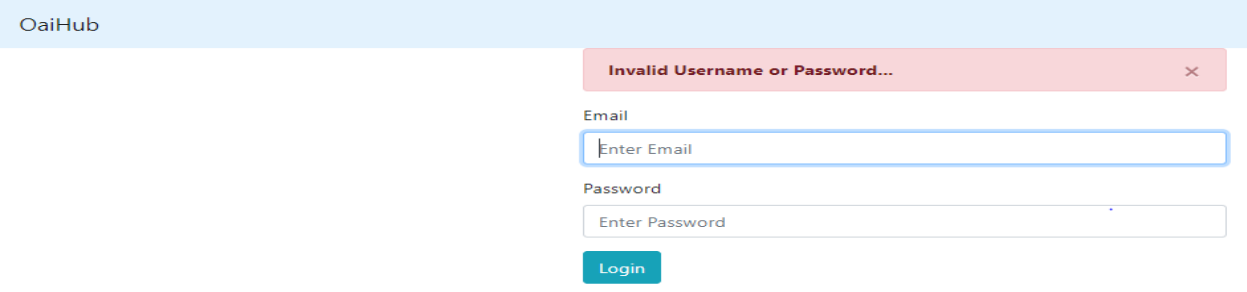
Chapter 10 – Testing

10.1 Unit Testing

Unit Test: 1		Test Class: User Login Window	Designed By	
Data Source: User input		**Objective:** Password checking test.	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for checking wrong password.	Enter login Info, User Email: Password:	Show message “Invalid username or Password”	Perfect

The test result screenshot for Unit Test 1 is given below,

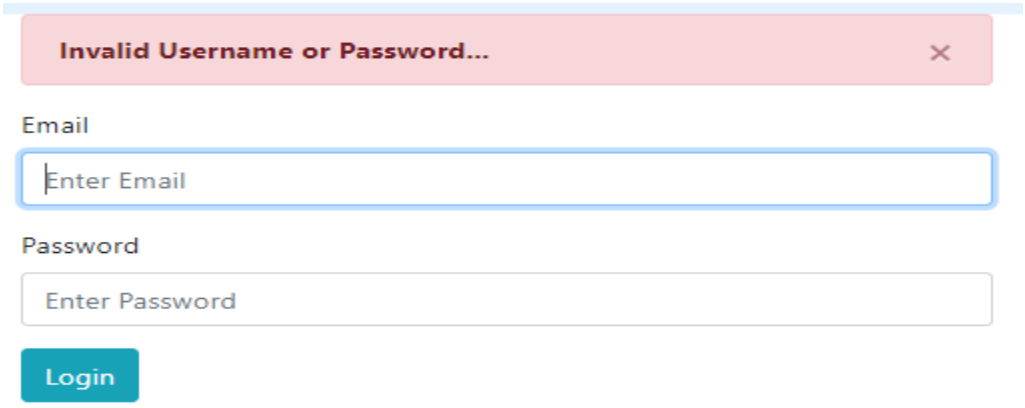




©Daffodil International University

Unit Test: 2		Test Class: Admin Login Window	Designed By	
Data Source: Admin input		**Objective:** Password checking test.	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for checking wrong password.	Enter login Info, Admin Email: Password:	Show message “Invalid username or Password”	Perfect

The test result screenshot for Unit Test 2 is given below,



©Daffodil International University

Unit Test: 3		Test Class: File	Designed By	
Data Source: File upload		**Objective:** Required field can't empty.	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.2	Test for Required field has been able to upload any file.	Enter file Info: Exam ID: select one File Type: select any File Name: Choose file Description:	Show message "Fail"	Perfect

The test result screenshot for Unit Test 3 is given below,

©Daffodil International University

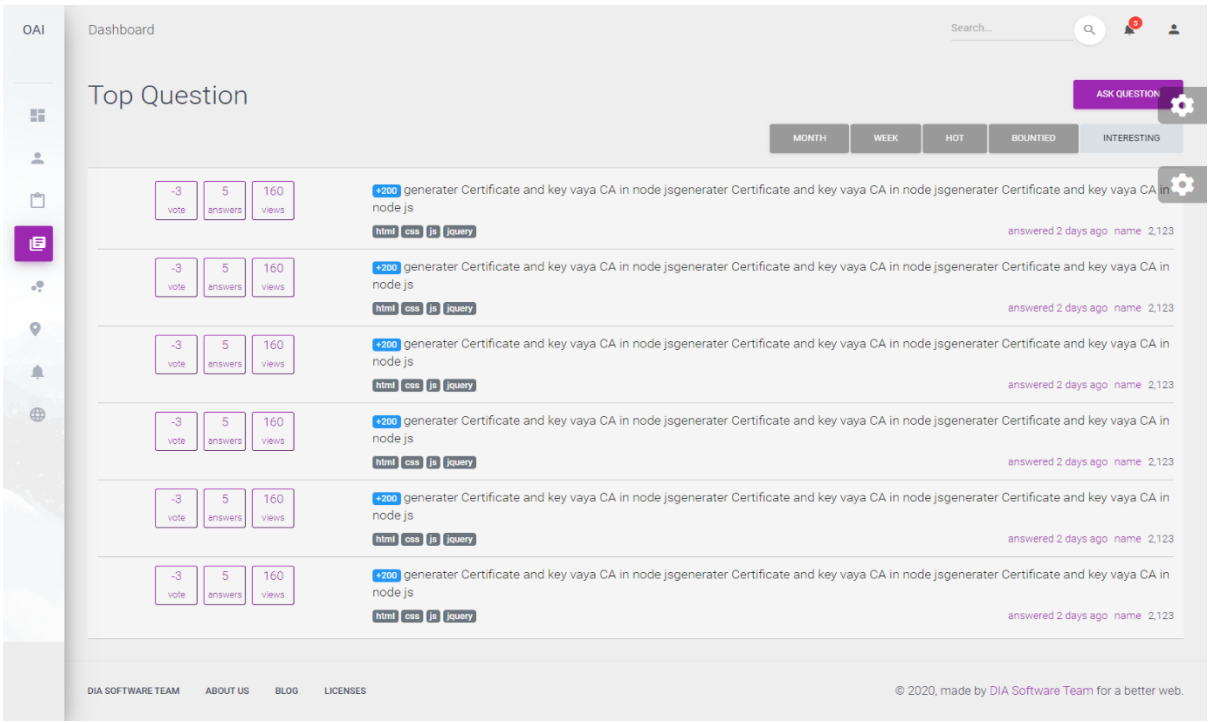
Forum voting action unit testing:

Test Priority: High	**Test Execute by:** DIA Intern Team
Unit test No: 01	**Test Execute Date:** 10/05/2020
Test case: Forum View	
Objective: The post came up properly on this page	
Data Source: What the user is posting	

Case No.	Description	Tasks	Result		Status (Pass/Fail)
			Actual result	**Expected result**	
1	We will check if the post is coming to this page properly	Post some question	All post show in the page	All post will be show in the page	Pass

The test result screenshot for Unit Test is given below

©Daffodil International University

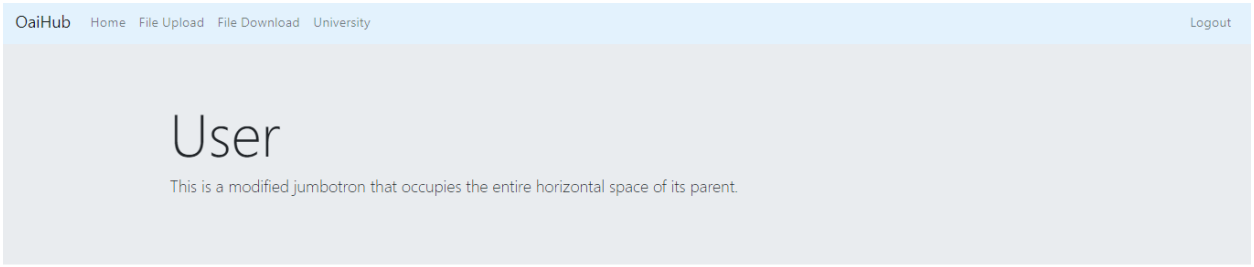


©Daffodil International University

10.2 Integration Testing:

Integration Test: 1		**Test Class: User**	**Designed By**	
Data Source: User input		**Objective:** Test for basic functionality.	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for basic function for user login	Enter Email: dity@g.com Enter password: 12345678	Show the user login page	Perfect

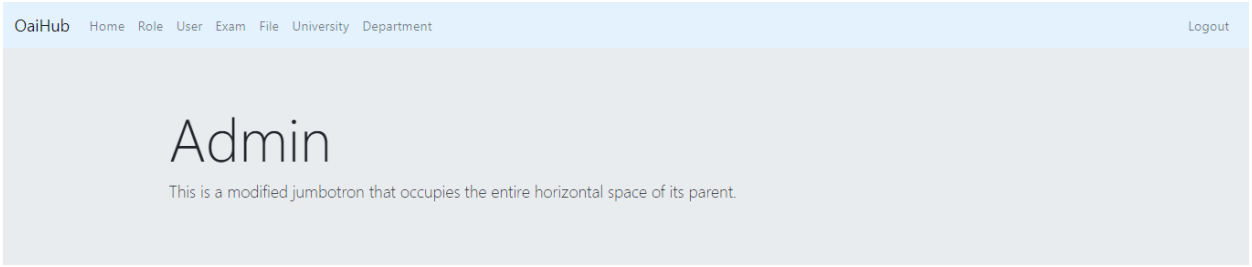
The test result screenshot for Integration Test 1 is given below,



©Daffodil International University

Integration Test: 2		Test Class: Admin	Designed By	
Data Source: Admin input		**Objective:** Test for basic functionality.	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for basic function for admin login	Enter Email: dsa@dsa.ds Enter password: 12345678	Show the admin login page	Perfect

The test result screenshot for Integration Test 1 is given below,

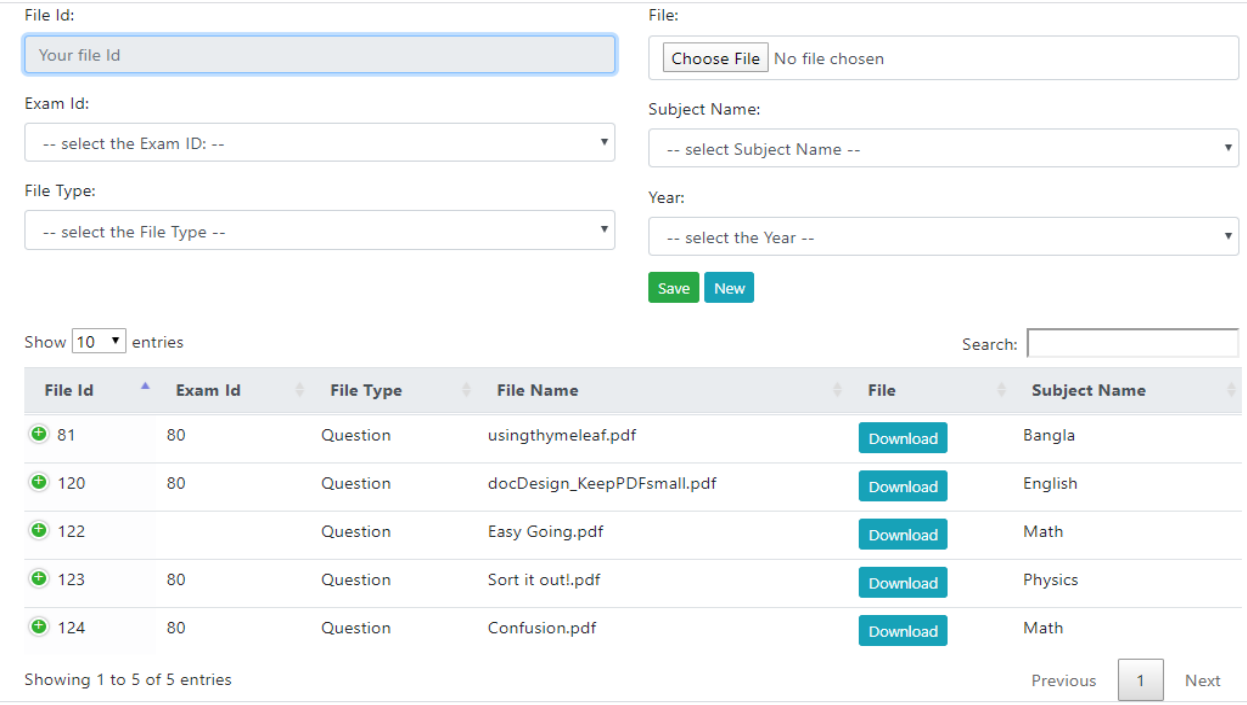


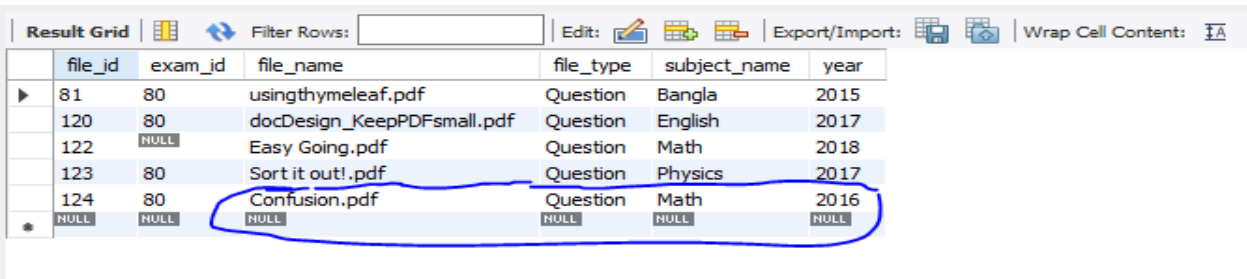
©Daffodil International University

Integration Test: 3		Test Class: File upload	Designed By	
Data Source: MySQL Database		**Objective:** Test for uploading any file	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for uploading any file	Save file: Click save button to save the file.	Show the message that your file uploaded successfully.	Perfect

©Daffodil International University

Integration Test: 4		Test Class: File upload	Designed By	
Data Source: MySQL Database		**Objective:** Test for show the uploaded file	**Tester:** Diainternteam	
Test Case	**Description**	**Tasks**	**Expected Result**	**Actual Result**
1.1	Test for uploading any file	Save file: Click save button to save the file.	Admin can show the uploaded file.	Perfect





©Daffodil International University

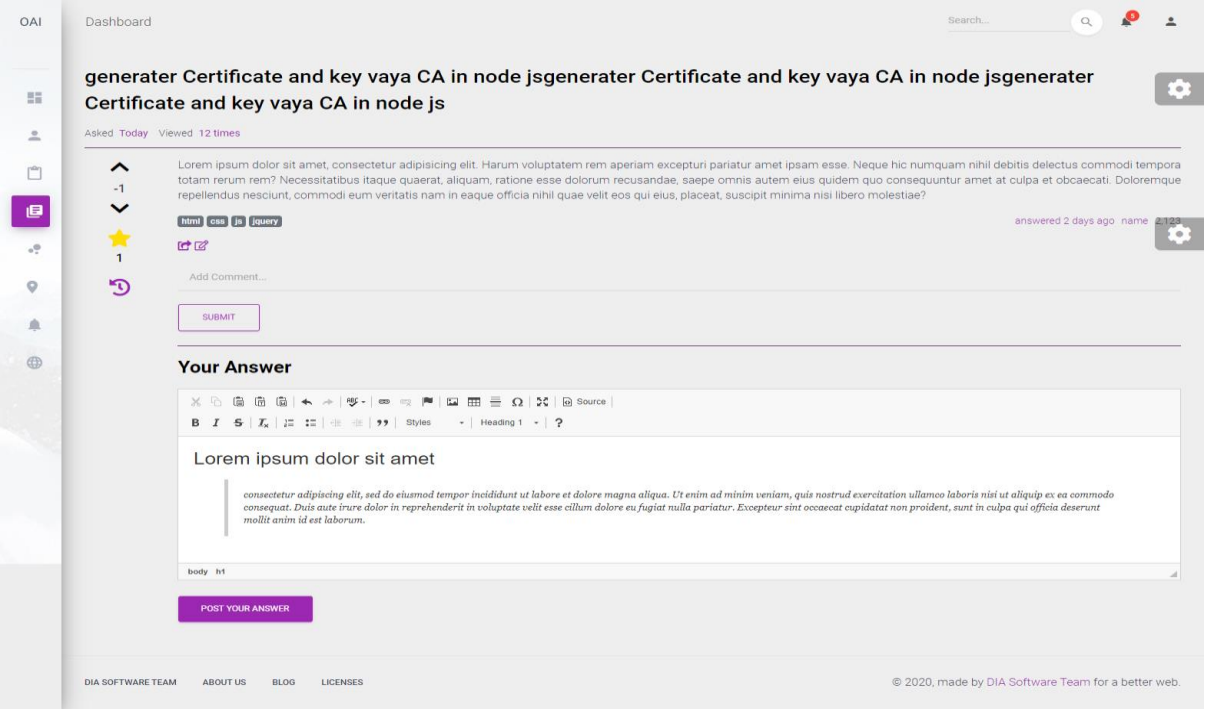
Forum voting action integration testing:

Test Priority: High	**Test Execute by:** DIA Intern Team
Integration test No: 03	**Test Execute Date:** 10/05/2020
Test case: Bookmark and voting	
Objective: Check bookmark and voting functionality	
Data Source: User Input	

Case No.	Description	Tasks	Result		Status (Pass/Fail)
			Actual result	**Expected result**	
1	We will check this functionality of bookmarking and voting here	Action by user click	Bookmark and voting successfully happened	Bookmark and voting will be successfully happened	Pass

©Daffodil International University

The test result screenshot for Integration Test is given below



©Daffodil International University

Chapter 11 – Implementation

11.1 Description

Basically, implementation assigns the user's handle of the organization's or institution’s workflow. The client endeavors to upgrade the coordinates work employing a structured method. Usage may be a procedure, strategy, or any plan, thought, demonstrate, portrayal, standard, or arrangement for undertaking something. Execution, as like, is the misuse that must track any early on considering so that something really happens. The extended group creates genuine development amid usage. Implementation of the innovation can be an energizing stage for the client since their extended thought gets to be something tangible. The extended designers begin building the program and coding it. Within the area, I will portray how I have executed everything approximately this preparation.

11.2 Training

Execution incorporates a grouping of exercises, through which preparing supervisors bring the course concurring to the affirmed plan to the learners. It requires planning of courses, resources, gear, and benefit suppliers aside from organizing continuous back for the classroom and guaranteeing the smooth stream of exercises as per the arrangement. An efficient, step-by-step handle process is utilized to construct a viable training program. Preparing activities that are standing alone regularly fall flat to meet organizational objectives and desires of the members.

11.3 Assess training needs

Identifying and evaluating needs is the first step to developing a training program. Training needs of employees may already be identified in strategic, human resources or individual development plans of the organization.

©Daffodil International University

11.4 Set organizational training objectives

Assessments of the training needs (administrative, task & separate) will identify any gaps in your current training initiatives and skill sets for employees. These gaps should be analyzed and prioritized, and transformed into the training goals of the organization. The ultimate aim is to bridge the gap between current performance and desired performance by developing a training program.

11.5 Create training action plan

The next step is to create a comprehensive action plan which includes theories of learning, instructional design, content, materials and any other elements of training. Methods for delivering resources and training should be detailed as well. The level of training and the learning styles of the participants also need to be considered when developing the program.

11.6 Implement training initiatives

The phase of implementation is where the workout program comes to life. Organizations need to choose whether in-house or outwardly coordinated training will be provided. Implementation of the program includes scheduling training activities and organizing any associated resources facilities, equipment, etc. Subsequently, the training program is officially launched, promoted and run.

11.7 Evaluate & revise training

As mentioned in the last segment, there should be continuous monitoring of the training program. Eventually, the entire program should be evaluated to regulate whether it was successful and meeting the training goals. Feedback from all stakeholders should be obtained in order to determine the effectiveness of the program and instructor and also knowledge or skill acquisition. (training-program, 2020)

©Daffodil International University

11.8 Big Bang

Big bang implementation on a single site is considerably easier to manage over multiple sites than a simultaneous big bang. The usually held view is that implementations with big bang have an inherently higher risk level. In one instance, implementation happens. On a given date, all users move onto the new system. The scope of a big bang implementation can also mean that it is problematic to accomplish complete end to end system testing and it is only when the system goes live that all interdependencies are fully tested.

Chapter 12 - Critical Appraisal and Evaluation

Within this chapter the project overview will be explained. In this topic both the success and system failure will be discussed. What's more, what experience we gained throughout the project will also be discussed. Likewise, it includes the success factor, the amount of objective it met, and what features it could not have met and the reason with the justification. In this section we will describe how I can meet my objective goal despite various obstacles.

12.1 Objective that could be met

The project proposed has met some objectives outlined below,

1. Implementing a platform for all student and guardian can get any information about any institution.
2. This application provides previous question bank with solutions, which can help the students and they can find these easily.
3. This tis the mature platform for both students and the educational organization.
4. Fix a methodology suitable for implementing a system.
5. Make the project well documented with maintenance the standard.
6. The application has to error free as much as possible.

©Daffodil International University

7. Messaging and commenting system were created for the user (student and guardian) to organization communication.
8. We are making this application's database which is based on this project requirement.
9. This application is web-based application which is portable easy to use from anywhere.
10. We are making this application with met the requirement which is given by our moderator.
11. We have made this system useful for different types of user and that will be very helpful for the user.

12.2 Success rate against each objective

Success rate to this objective is relatively satisfactory. This rate is impartially alright of this objective as both student/guardian and organizations can be able to get their own expected outcome. The organizations or institutions can share their information and student can view their descriptions and get their expected information's this thing will help reduce the hassle of the students also guardian. And it will help more students to collect questions from different years. As a student and as a guardian they will get exactly the information they need from here.

12.3 How much better it could be done

We are following many structures to standardize this web application. There were many diagrams in these diagrams that helped us a lot to maintain our sequence like as activity diagram, sequence diagram, ERD diagram, use case diagram and class diagram etc. At the same time, we have tried to do proper documentation of all the diagrams which will help to do more with it in the future. There are some diagrams that we could not add to this documentation because of the security of the organization. And there are a lot of

©Daffodil International University

options here that have been made for the student or organizational user in a very convenient way to understand.

12.4 How better are the features of the solution?

This web application might be more friendly to the users. If we want to show as an example then we will see that the forum part has been made very interactive. Any student or user can ask his/her questions at any moment. And there is a facility to comment here, if anyone knows the answer, it can be informed through comments which is very much responsive. As a result, it can solve any types of problem very quickly and takes very close to the solution. If the time of the project is extended a bit this system is better than this. The workload was so high, though it was teamwork.

Chapter 13 - Conclusion

13.1 Conclusion

For developing this project ‘OAIHUB’ I have tried heart and soul. The overview of the entire documentation contains the goals and the success inside this section. This section also defines implementing knowledges and project values. I have faced many problems during the development and our mentor help us to overcome the situation. Many more features are not done yet for the short time of the project that will be developed in the next. Here I outlined my summary of the total work such as the main goal, my experience, project value and lots of things. The full project will be very much benefitted for the students. It has so many details those are given below,

13.2 Summary of the project

By following the task, I have appropriately instigated all the project requirements and its works perfectly. I worked on this project for about six months, during this time I have implemented many things in this project. I have worked in many fields while doing this project. I had to collect a lot of in-depth data from various organizations and students and this core data is working very well in our system. For this we did a very large survey.

We have done the engineering part of this system very nicely which can be understood by looking at our documents. We have described all kinds of things in a very stunning

©Daffodil International University

way here such as software architecture design, literature analysis, using different methodology, diagrams etc. Here the project standards are measured by means of different categories of assessment techniques. I can say that this documentation does all categories of project stuff. I have implemented this system by succeeding those analyzes and trying to make the system which the fixe the real-world problem. And which will be of great benefit to the students in the future as well as to the benefit of various educational organizations.

13.3 Goal of the project

We have created this web application for different students and different educational institutions. We have tried to fulfill all the requirements and I am describing the goal which I have identified in below,

- Using this system, the user can get any information about any institution.
- The previous online question bank can help the students. They can find these easily.
- Students can know the admission procedure in any school, college or university.
- Students will be able to attend various online exams like (IELTS) which they will be able to attend through the pro feature.
- It is an educational web application that will help students in a variety of ways.
- With this we have created a communication system and a system for taking feedback from the user.
- User can discuss anything in the user forum.

13.4 Success of the project

The success of each project depends on the acceptance of its respective users and we've been very successful in bringing it to our users. I would like to say that we have done all these fields to fulfill the requirements of this project. We have met all the objectives with data from different students and educational institutions. We have developed a forum through which any student and any user can find a solution to a problem very quickly and

©Daffodil International University

this forum is very interactive and user friendly for users of every level. Lastly, I can say that we have been able to fulfill all the objectives in a very efficient way and that is the main success of our project.

13.5 Value of the project

The necessities of our life are increasing day by day, and our problems are increasing day by day along with our needs. The students of A level and O level in our country need a lot of information to get admission in different universities at some stage of their education life, not only do i have to talk to our students but the parents we have also need a lot of information to enroll their children in different schools and colleges. We have built-up this web application keeping their words in mind. All the information for admission of a student and a parent through the application can be found here which will be very easy for them. I have created a pro user facilities and voting system in forum here through which any student can gain knowledge by asking questions on any of his topics. Which will bring a lot of good for our country.

13.6 My experience

I learned a lot of things during the internship that was very new to me. I have worked on a web applications name "**OAIHUB"**. When I work on it, I learn a lot about lots of things, such as database design, database architecture also I've done a lot of work on how to connect the back-end with the front-end. I have developed **"OAIHUB"** web application using the java Spring Boot Framework. I learn a lot of new topics, new skills I can add as my experience. The biggest thing I have learned is that if I run into a problem, I will find a way to solve it. I have become skilled at the problem solving very well. The thing that I have learned very well from here is that one has to finish a job completely despite the pressure. The requirements that I was able to fulfill in a very efficient way through teamwork. From here I learned how to manage a team and work with the team and how to solve problem by working together. I believe that this experience will be very useful in my future life.

©Daffodil International University

References

Anon., 2019. *scruminc.* [Online]
Available at: https://www.scruminc.com/what-is-timeboxing/
[Accessed 2020].

Anon., 2020. *guru99.* [Online]
Available at: https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html
[Accessed 2020].

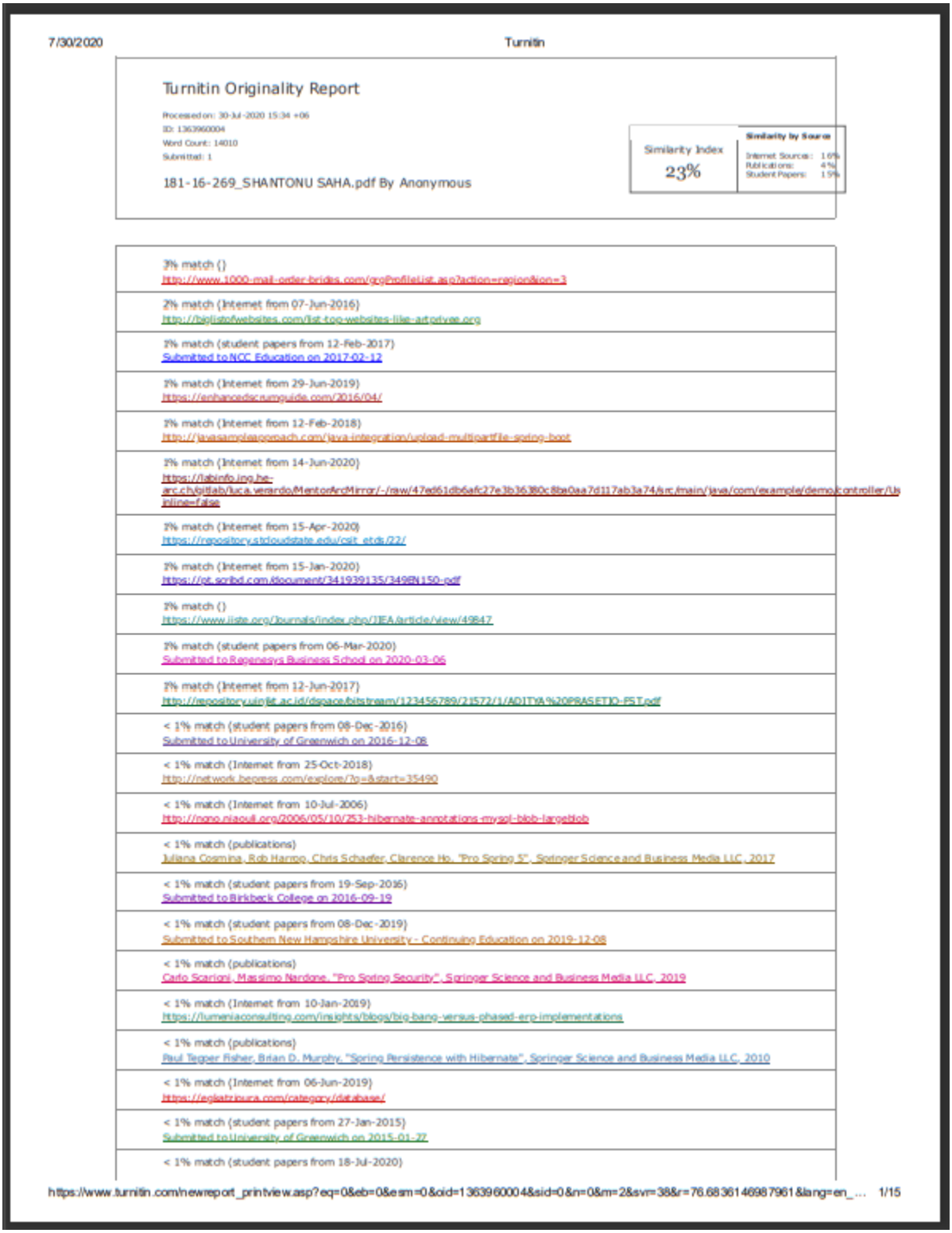
Anon., 2020. *softwaretesting fundamentals.* [Online]
Available at: http://softwaretestingfundamentals.com/unit-testing/#:~:text=UNIT%20TESTING%20is%20a%20level,and%20usually%20a%20single%20output.
[Accessed 2020].

https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html, 2020. test plan.

training-program, 2020. *https://explorance.com/.* [Online]
Available at: https://explorance.com/blog/5-steps-to-creating-effective-training-programs/

©Daffodil International University

Plagiarism report:



7/30/2020 Turnitin

Turnitin Originality Report

Processed on: 30-Jul-2020 15:34 +06
ID: 1363960004
Word Count: 14010
Submitted: 1

181-16-269_SHANTONU SAHA.pdf By Anonymous

Similarity Index 23%

Similarity by Source
Internet Sources: 16%
Publications: 4%
Student Papers: 15%

3% match ()
http://www.1000-mail-order-brides.com/org/profileList.asp?action=region&lon=3

2% match (Internet from 07-Jun-2016)
http://biglistofwebsites.com/list-top-websites-like-artofvae.org

1% match (student papers from 12-Feb-2017)
Submitted to NCC Education on 2017-02-12

1% match (Internet from 29-Jun-2019)
https://enhancediscussguide.com/2016/04/

1% match (Internet from 12-Feb-2018)
http://javasampleapproach.com/java-integration/upload-multipartfile-spring-boot

1% match (Internet from 14-Jun-2020)
https://labinfo.ing.he-arc.ch/ljdfab/luca.verardo/MentorArcMirror/-/raw/47ed61db6afc27e3b36380c8be0aa7d117ab3a74/src/main/java/com/example/demo/controller/Us inline=false

1% match (Internet from 15-Apr-2020)
https://repository.stcloudstate.edu/csit_etds/22/

1% match (Internet from 15-Jan-2020)
https://pt.scribd.com/document/341939135/349BN150-pdf

1% match ()
https://www.iiste.org/Journals/index.php/JIEA/article/view/49847

1% match (student papers from 06-Mar-2020)
Submitted to Regenesys Business School on 2020-03-06

1% match (Internet from 12-Jun-2017)
http://repository.uinjkt.ac.id/dspace/bitstream/123456789/21572/1/AD1TYA%20PRASETIO-FST.pdf

< 1% match (student papers from 08-Dec-2016)
Submitted to University of Greenwich on 2016-12-08

< 1% match (Internet from 25-Oct-2018)
http://network.begeess.com/explore/?g=&start=35490

< 1% match (Internet from 10-Jul-2006)
http://nono.ma.cui.org/2006/05/10/253-hibernate-annotations-mysql-blob-largeblob

< 1% match (publications)
Juliana Cosmina, Rob Harrop, Chris Schaefer, Clarence Ho. "Pro Spring 5", Springer Science and Business Media LLC, 2017

< 1% match (student papers from 19-Sep-2016)
Submitted to Birkbeck College on 2016-09-19

< 1% match (student papers from 08-Dec-2019)
Submitted to Southern New Hampshire University - Continuing Education on 2019-12-08

< 1% match (publications)
Carlo Scarioni, Massimo Nardone. "Pro Spring Security", Springer Science and Business Media LLC, 2019

< 1% match (Internet from 10-Jan-2019)
https://lumeniaconsulting.com/insights/blogs/big-bang-versus-phased-erp-implementations

< 1% match (publications)
Paul Tepper Fisher, Brian D. Murphy. "Spring Persistence with Hibernate", Springer Science and Business Media LLC, 2010

< 1% match (Internet from 06-Jun-2019)
https://eglabtripura.com/category/database/

< 1% match (student papers from 27-Jan-2015)
Submitted to University of Greenwich on 2015-01-27

< 1% match (student papers from 18-Jul-2020)

https://www.turnitin.com/newreport_printview.asp?eq=0&eb=0&esm=0&oid=1363960004&sid=0&n=0&m=2&svr=38&r=76.68361469879 61&lang=en_... 1/15

©Daffodil International University