**Project Name: Institute Management Portal**

**Project Member:**

**Kishori Attarde 220343120048**

**Aditya Deshpande 220343120032**

**Aishwarya Shinde 220343120101**

**Swapnil Pachupate 220343120106**

**Abstract:**

Over the years, portals have become popular in the information system community. A web portal is seen as a special internet (or intranet) site designed to act as a gateway to give access to other sites. A portal aggregates information from multiple sources and makes that information available to various users. Portals were projected to turn the web from institution-centric repository of information to a dynamic user-centric collection of everything useful to the person in a particular role and this is projection has been realized. Portals have these days become of interest in higher-education institutions. Institute portals integrate campus-specific information and activities which is stored in the campus electronic vaults such as databases, file systems and existing application systems, with unstructured data (text) from on and off campus. Portal technology is used as a tool to enhance institutional efficiency and productivity.

The system is implemented using a 3-tier approach, with a backend database, a middle tier of Spring Boot MVC, and web browser as the front-end client.

In order to develop a portal system, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client-side scripting techniques, implementation technologies such as Spring Boot MVC, programming language (such as Core Java, Advance Java), relational/non-relational databases (such as MySQL).

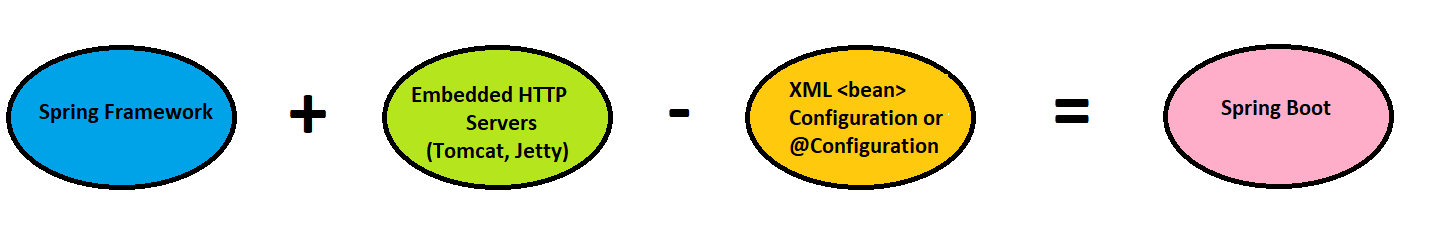
The main objective behind construction of this portal is to provide a single place to students from where they can do all the study related activates without login to different websites. In other words, to provide a single place for all kind of jobs. Main objective of proposed system is not only to be useful to students of the college but also staff members & administrative persons of the college. System objective is of being useful in significant way by providing most basic & most essential functionalities & features to its users in efficient & effective manner.

**Implementation Technologies:**

1. **Spring Framework:**

Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.

It is a Spring module that provides the **RAD (***Rapid Application Development***)** feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.



In short, Spring Boot is the combination of **Spring Framework** and **Embedded Servers**.

In Spring Boot, there is no requirement for XML configuration (deployment descriptor). It uses convention over configuration software design paradigm that means it decreases the effort of the developer.

We can use Spring **STS IDE** or **Spring Initializr** to develop Spring Boot Java applications.

**1.1 Features of Spring Framework:**

* Create stand-alone Spring applications
* Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
* Provide opinionated 'starter' dependencies to simplify your build configuration
* Automatically configure Spring and 3rd party libraries whenever possible
* Provide production-ready features such as metrics, health checks, and externalized configuration
* Absolutely no code generation and no requirement for XML configuration

**1.2 Advantages of Spring Boot Framework:**

* It creates stand-alone Spring applications that can be started using Java -jar.
* It tests web applications easily with the help of different Embedded HTTP servers such as Tomcat, Jetty, etc. We don't need to deploy WAR files.
* It provides opinionated 'starter' POMs to simplify our Maven configuration.
* It provides production-ready features such as metrics, health checks and externalized configuration.
* There is no requirement for XML configuration.
* It offers a CLI tool for developing and testing the Spring Boot application.
* It offers the number of plug-ins.
* It also minimizes writing multiple boilerplate codes (the code that has to be included in many places with little or no alteration), XML configuration, and annotations.
* It increases productivity and reduces development time.

1. **MongoDB**

MongoDB is an open source [NoSQL](https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL) database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

* 1. **Features of MySQL:**
* **Ad-hoc queries for optimized, real-time analytics**

When designing the schema of a database, it is impossible to know in advance all the queries that will be performed by end users. An ad hoc query is a short-lived command whose value depends on a variable. Each time an ad hoc query is executed, the result may be different, depending on the variables in question. MongoDB supports field queries, range queries, and regular expression searches. Queries can return specific fields and also account for user-defined functions. This is made possible because MongoDB indexes BSON documents and uses the MongoDB Query Language (MQL).

* **Indexing appropriately for better query executions**

Without the right indices, a database is forced to scan documents one by one to identify the ones that match the query statement. But if an appropriate index exists for each query, user requests can be optimally executed by the server. MongoDB offers a broad range of indices and features with language-specific sort orders that support complex access patterns to datasets. Notably, MongoDB indices can be created on demand to accommodate real-time, ever-changing query patterns and application requirements.

* **Replication for better data availability and stability**

When your data only resides in a single database, it is exposed to multiple potential points of failure, such as a server crash, service interruptions, or even good old hardware failure. In MongoDB, replica sets are employed for this purpose. A primary server or node accepts all write operations and applies those same operations across secondary servers, replicating the data. If the primary server should ever experience a critical failure, any one of the secondary servers can be elected to become the new primary node. And if the former primary node comes back online, it does so as a secondary server for the new primary node.

* **Sharding**

When dealing with particularly large datasets, sharding—the process of splitting larger datasets across multiple distributed collections, or “shards”—helps the database distribute and better execute what might otherwise be problematic and cumbersome queries. Without sharding, scaling a growing web application with millions of daily users is nearly impossible.

Like replication via replication sets, sharding in MongoDB allows for much greater horizontal scalability. Horizontal scaling means that each shard in every cluster houses a portion of the dataset in question, essentially functioning as a separate database. The collection of distributed server shards forms a single, comprehensive database much better suited to handling the needs of a popular, growing application with zero downtime.

* **Load Balancing**

At the end of the day, optimal load balancing remains one of the holy grails of large-scale database management for growing enterprise applications. Fortunately, via horizontal scaling features like replication and sharding, MongoDB supports large-scale load balancing. The platform can handle multiple concurrent read and write requests for the same data with best-in-class concurrency control and locking protocols that ensure data consistency. There’s no need to add an external load balancer—MongoDB ensures that each and every user has a consistent view and quality experience with the data they need to access.

1. **ReactJS**

ReactJS is a **declarative**, **efficient**, and flexible **JavaScript library** for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.

**3.1 Why Use ReactJS?**

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

**3.2 Features**

* **JSX**

JSX stands for JavaScript XML. It is a JavaScript syntax extension. Its an XML or HTML like syntax used by ReactJS. This syntax is processed into JavaScript calls of React Framework.

* **Components**

ReactJS is all about components. ReactJS application is made up of multiple components, and each component has its own logic and controls. These components can be reusable which help you to maintain the code when working on larger scale projects.

* **One-way Data Binding**

ReactJS is designed in such a manner that follows unidirectional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application. If the data flow is in another direction, then it requires additional features.

* **Virtual DOM**

A virtual DOM object is a representation of the original DOM object. It works like a one-way data binding. Whenever any modifications happen in the web application, the entire UI is re-rendered in virtual DOM representation. Then it checks the difference between the previous DOM representation and new DOM.

* **Simplicity**

ReactJS uses JSX file which makes the application simple and to code as well as understand. We know that ReactJS is a component-based approach which makes the code reusable as your need. This makes it simple to use and learn.

* **Performance**

ReactJS is known to be a great performer. This feature makes it much better than other frameworks out there today. The reason behind this is that it manages a virtual DOM. The DOM is a cross-platform and programming API which deals with HTML, XML or XHTML. The DOM exists entirely in memory. Due to this, when we create a component, we did not write directly to the DOM. Instead, we are writing virtual components that will turn into the DOM leading to smoother and faster performance.

**3.3 Advantages of ReactJS**

* **Easy to Learn and Use**
* **Creating Dynamic Web Applications Becomes Easier**
* **Reusable Components**
* **Performance Enhancement**
* **The Support of Handy Tools**
* **Known to be SEO Friendly**
* **The Benefit of Having JavaScript Library**
* **Scope for Testing the Codes**

1. **Hardware and Software Requirements (Minimum):**

**Hardware:**

1. Intel i3 processor 3rd generation or later / AMD Ryzen 200 2nd generation or later

2. 2 GB ddr3 ram.

3. Windows 7 Home edition or later.

4. 200 GB Data HDD Space

5. Data Connection 200 kbps

**Software:**

1. Eclipse 4.7.3 Oxygen
2. MongoDB 5.0
3. Google Chrome version 105.0
4. Apache Tomcat Server 8.5
5. Maven Dependencies
6. **ER Diagram:**

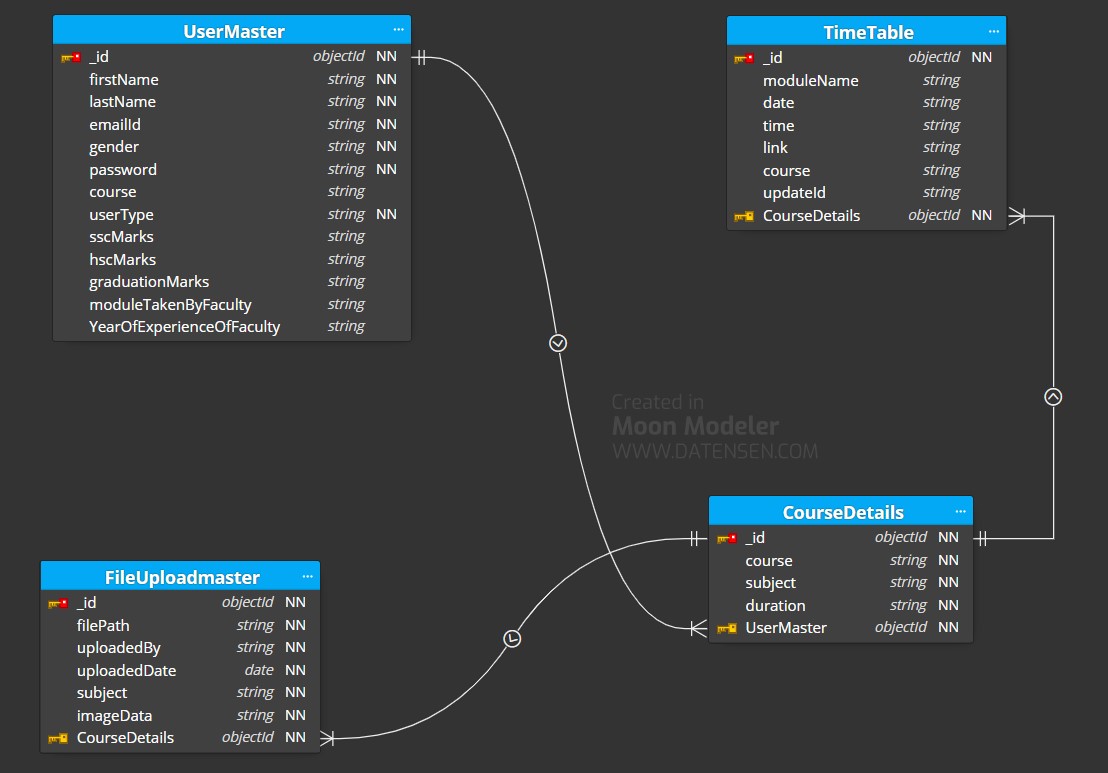


Figure 1: ER Diagram

1. **Table Structures:**
2. **Collection name: UserMaster**

|  |  |
| --- | --- |
| **Column Name** | **Type** |
| Id | String |
| firstName | String |
| lastName | String |
| emailId | String |
| gender | String |
| password | String |
| course | String |
| userType | String |
| hscMarks | String |
| sscMarks | String |
| graduationMarks | String |
| moduleTakenByFaculty | String |
| QulificationOfFaculty | String |
| YearOfExperienceOfFaculty | String |

1. **Collection name: CourseDetails**

|  |  |
| --- | --- |
| **Column Name** | **Type** |
| courseId | String |
| course | String |
| subject | String |
| duration | String |

1. **Collection name: FileUploadMaster**

|  |  |
| --- | --- |
| **Column Name** | **Type** |
| Id | String |
| filePath | String |
| uploadedBy | String |
| uploadedDate | String |
| subject | String |
| imageData | String |

1. **Collection name: TimeTableMaster**

|  |  |
| --- | --- |
| **Column Name** | **Type** |
| Id | String |
| moduleName | String |
| Date | String |
| Time | String |
| Link | String |
| updateId | String |
| course | String |

1. **UML Diagrams:**

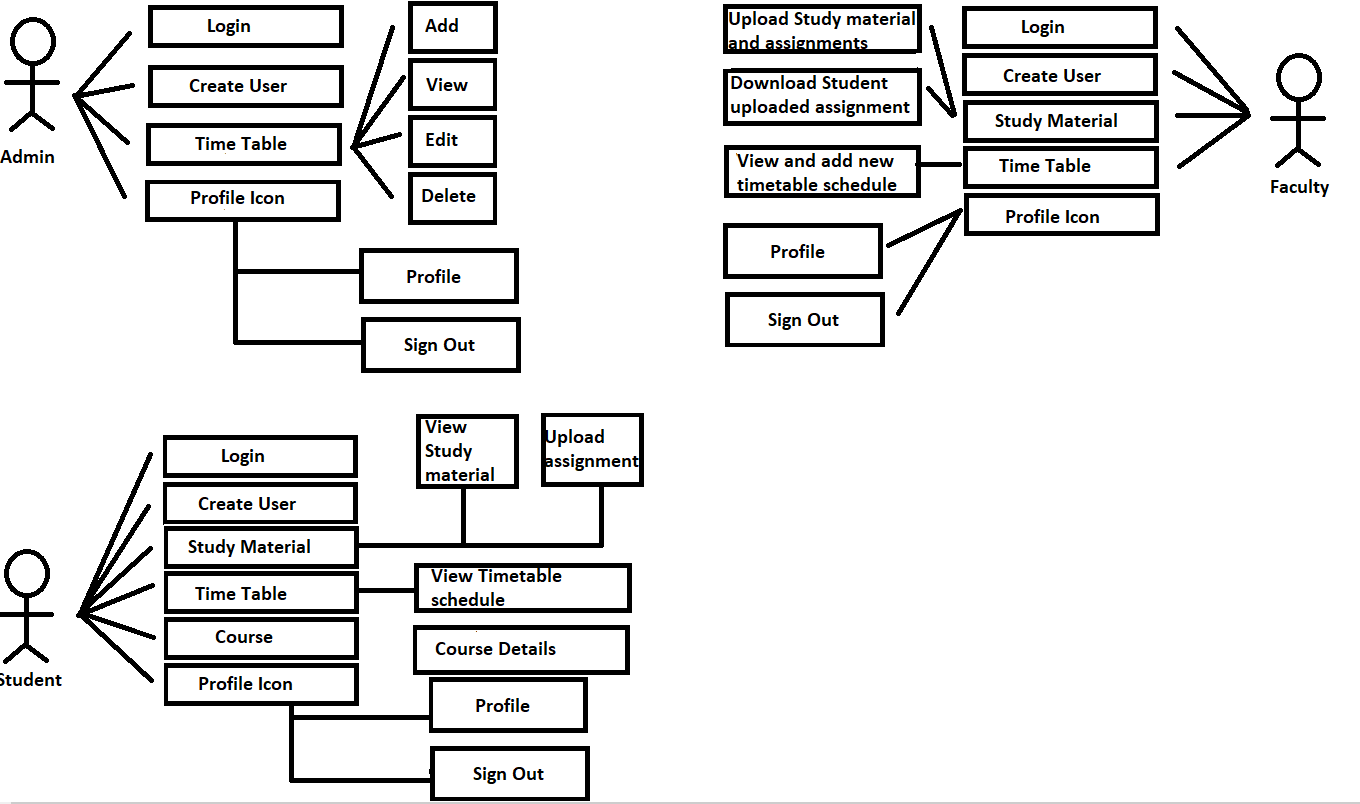


Figure 2: Use Case

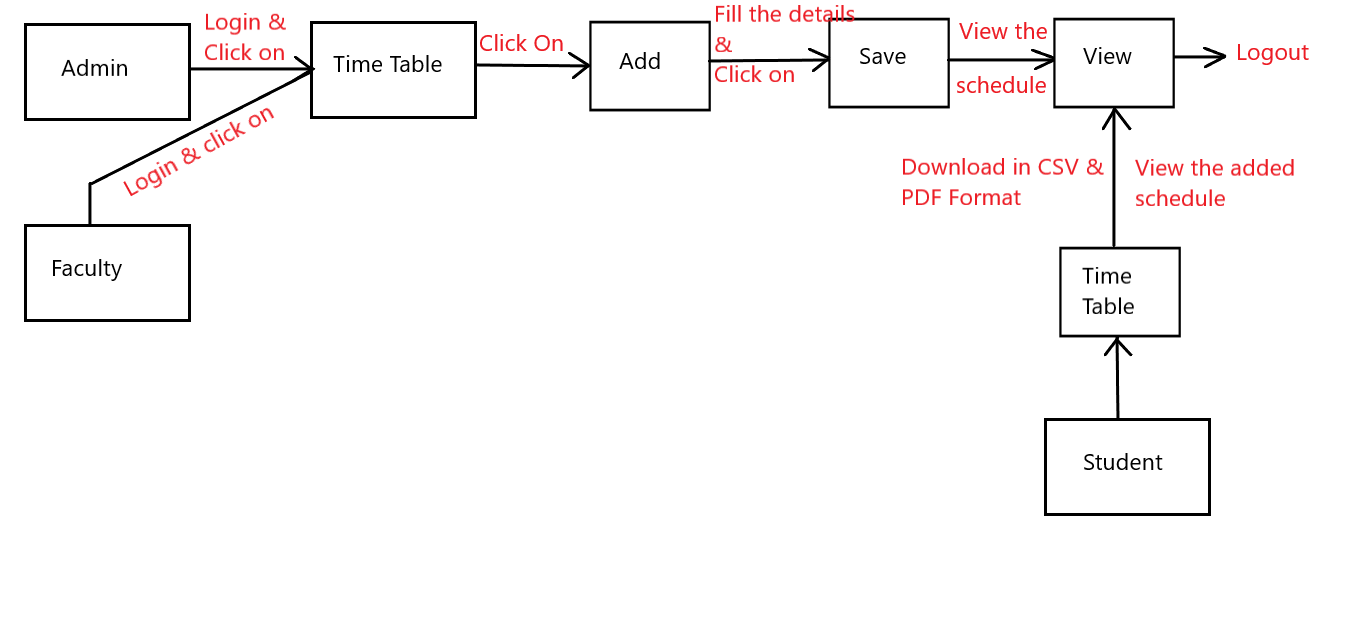


Figure 3: Collaboration Diagram

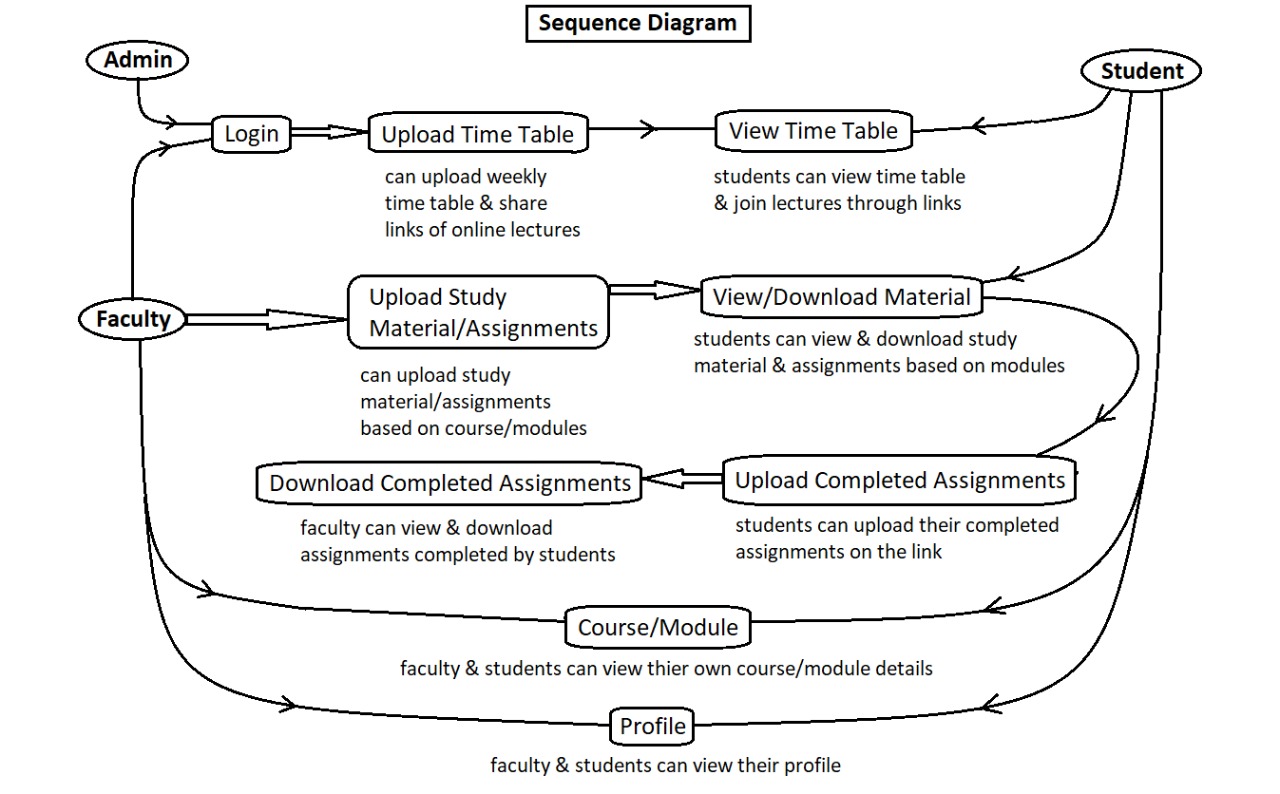


Figure 4: Sequence Diagram

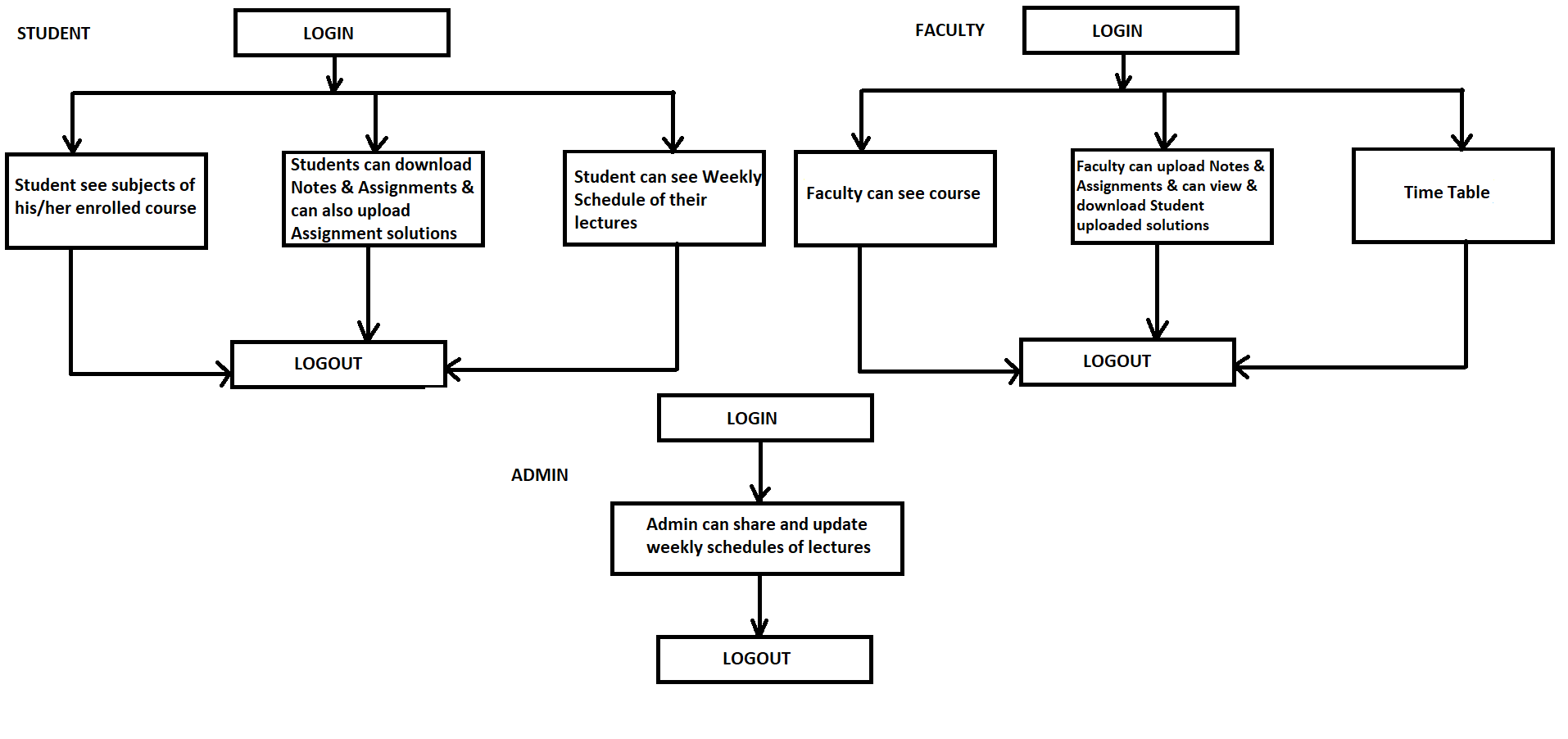


Figure 5: Component Diagram

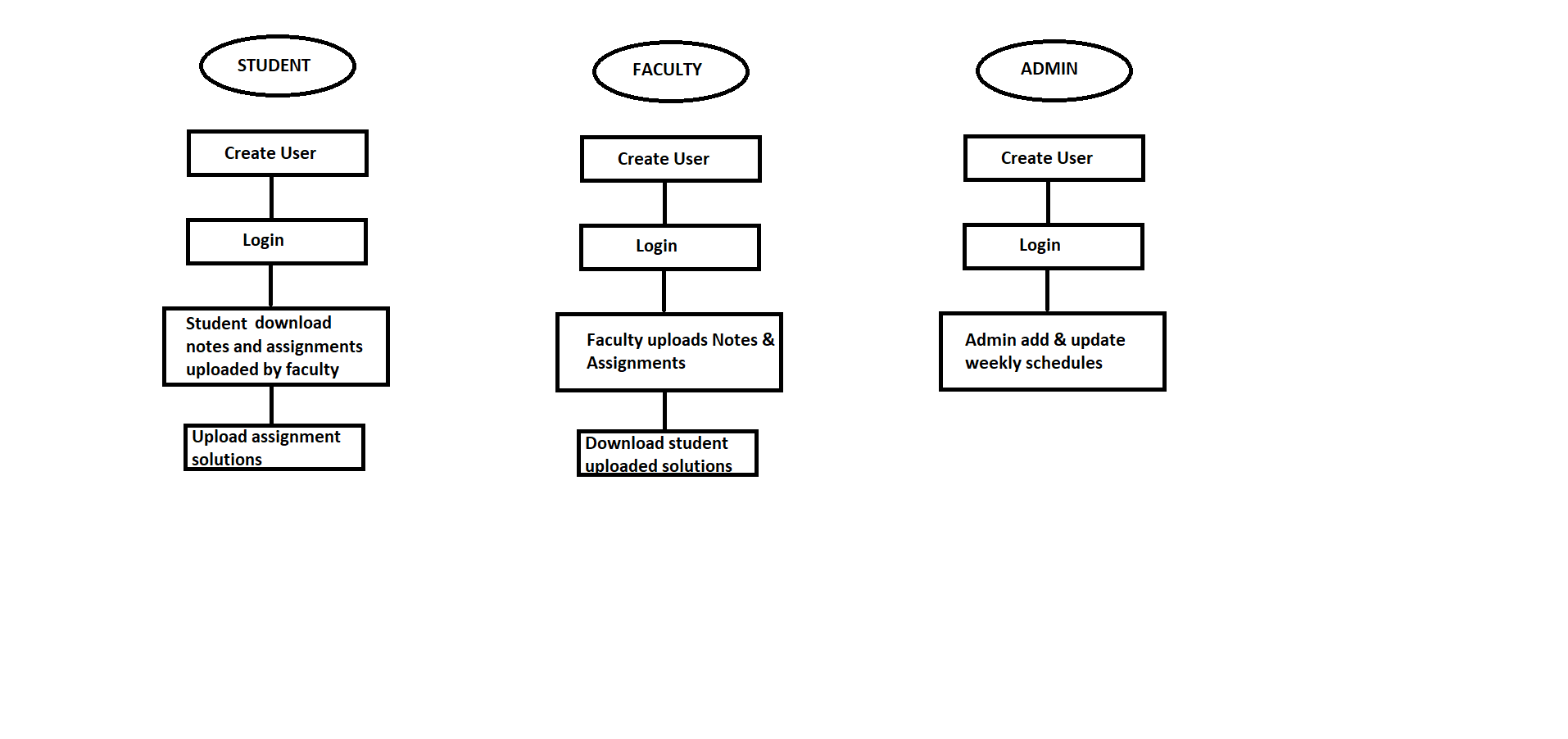


Figure 6: State Diagram

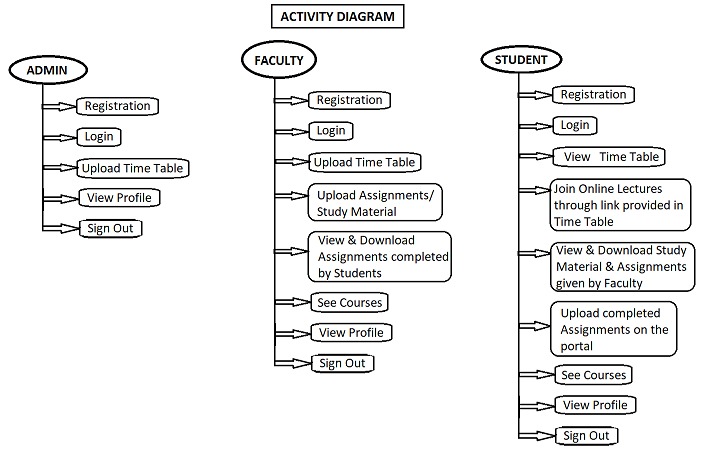


Figure 7: Activity Diagram

1. **End to End Flow of Application:**

**Student:**

* 1. Student will login to the portal or will register if he/she is not a registered user.
  2. After registration User will login in the portal.
  3. It will display Study Material, TimeTable and Course Tab.
  4. In Course Tab, Student will be able to view his/her enrolled course details.
  5. In Timetable tab, Student will be able to view his/her weekly schedule.
  6. Student can download the time table in csv and pdf format.
  7. Student can filter the data either by Module name, Date or Time.
  8. In Study Material tab, Student will be able to view module related notes shared by faculty as well as Assignment given if any.
  9. Student can download the uploaded material by selecting the module name and clicking on the file name.
  10. Student can upload completed assignment by clicking on upload button.
  11. Student will be displayed Profile tab where Student can view and edit details.

**Faculty:**

1. Faculty will login to the portal or will register if he/she is not a registered user.
2. After Registration, User will login into the portal.
3. Faculty will be displayed Study Material and Timetable tab
4. In Study Material Tab, Faculty can upload module related notes and Assignments.
5. Faculty can download student uploaded completed assignment from this tab only.
6. In Timetable Tab, Faculty will be able to view weekly schedules.
7. Faculty can add new timetable schedule as well by adding module name, date, time and lecture link for the same.
8. Faculty can download the time table in csv and pdf format.
9. Faculty can filter the data either by Module name, Date or Time.
10. Faculty will be displayed Profile tab where Faculty can view and edit details.

**Admin:**

* 1. Admin will login to the portal or will register if he/she is not a registered user.
  2. After registration, User will login into the portal.
  3. Admin will be displayed Timetable tab.
  4. Admin can add new timetable schedule as well by adding module name, date, time and lecture link for the same.
  5. Admin can download the time table in csv and pdf format.
  6. Admin can filter the data either by Module name, Date or Time.
  7. Admin will be displayed profile tab, where Admin can view and edit details.

1. **Future Scope of Project**
2. Mobile Application
3. Fees Payment from portal
4. Discussion Forum
5. Chatting System

**Thank You!**