**UCS1617 – MINI PROJECT**

**TEAM MEMBERS:**

**Sukeerthi M - 185001175**

**Vikraman S - 185001195**

**Yeshwanth S V - 185001204**

**Exam Registration System – ERS**

**PROBLEM STATEMENT FOR MINI PROJECT**

In the fast-moving world where technology is being used to replace conventional methods, one of the most useful systems would be to automate the process of registering for exams. It also provides a hassle-free way to verify the candidate’s identity and distribute hall tickets without having to endure the meticulous work load of mailing to each address. This system aims at minimizing the work load and use the time allocated to scheduling and managing a lot of resources into other effective strategies for smooth conduct of examinations.

Through the software, it would be effective to gather the details of the students that would be necessary for them to take up the examination without having to be physically present to authenticate their identity. Especially during the time of a pandemic like the current one, where it would be desirable for a non-contact mode for registration of examinations, the remote software could achieve the same.

The ideal situation for exam registration is that it should not be tiresome. There have been instances where people have given up going through the offline procedure due to health risks. In turn, it is possible that they are not able to make the most of their opportunities. In order to develop a solution that could be deployed in the real world, the online software for exam registration with credentials could be the most effective way to put the ideal circumstances in perspective for students and the examination process in general.

**Software Requirements Specification**

1. **Introduction**
   1. Purpose
   2. Scope
   3. Definitions, acronyms, and abbreviations
   4. References
   5. Overview
2. **Overall description**
   1. Product perspective
   2. Product functions
   3. User characteristics
   4. Constraints
   5. Assumptions and dependencies
3. **Specific requirements**

3.1. Non-functional Requirements

3.1.1. Software Used

3.1.2. Hardware Used

**1) Introduction**

The process of registering for examinations has been involved with filling an application and then manually entering the details of the student into the database and printing out the hall tickets in bulk followed by handing them over to the students. The Exam Registration system is used in the effective dispatch and easy access to all of the students. This system adopts a comprehensive approach to minimize the manual work and lays out the foundation for modernizing education rather than sticking with the conventional methods. Physical presence of the candidate would not be required and the identity verification can be carried out remotely by uploading scanned copies of documents. This reduces the time and resources of the students and helps the administrators to manage the secured documents in a safe and secured environment. It plays the role of a robust interface between the Student and the Exam Controller responsible for the Issue of Hall Ticket.

**1.1) Purpose**

The process of ensuring that the right candidates are able to take up the exam is getting cumbersome as the registrations are extensively increasing each year. If we limit ourselves to the conventional method of issuing the hall tickets, it could cause a lot of inconvenience for the people living in far-off places to collect the hall ticket and mark their presence whenever required. An automated system becomes essential to meet the demand of a hassle-free exam registration process and issue of hall tickets in time. This system thus makes sure to use appropriate programming and database techniques to elucidate the steps involved with satisfactory end user experience.

**1.2) Scope**

* The exam registration system provides a user-friendly online interface which enables them to register their identification details and submit the necessary documents like scanned copies of aadhar card to verify their genuinity.
* The controller who is responsible for the conducting of the examination has most of the workload minimized and acquires the convenience to process each application quickly and with more scrutiny.
* The system also functions as a middleware interface between the user and the exam controller to avoid the aspects of being physically present in order to clarify any queries regarding the registration process.
* The students who have enrolled for the examination will have transparent access to the status of the application and the scheduled dates for registration and uploading documents as well as releasing hall-tickets.

**1.3) Definitions, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Object** | **Explanation** |
| **Exam Controller** | Refers to the central authority who is the planner for the steps required to be fulfilled to successfully register for the examination. |
| **Student** | Refers to the one who wishes to register for the examination and obtain the hall-ticket. |
| **HTML** | Hypertext Markup Language used for creating web pages. |
| **ERS** | Exam Registration System. |
| **HTTP** | Hypertext Transfer Protocol. |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol is the communication protocol used to connect hosts on the Internet. |
| **GUI** | Graphical User Interface. |

**1.4) References**

Reference Websites

* <https://netbeans.org/kb/docs/java/quickstart.html>
* <https://www.w3schools.com/html/default.asp>
* <https://www.w3schools.com/css/default.asp>

**1.5) Overview**

SRS includes two sections overall description and specific requirements. The Overall Description will describe the major role of the system components and interconnections. The Specific Requirements will describe roles & functions of the actors.

**2) Overall Description**

The exam registration system encompasses various GUI interfaces at each stage of the registration process. The system consists of a database of student details and scanned copies of identity verification documents. A repetitive algorithm is used to generate a digitally verified and downloadable hall-ticket.

**2.1) Product Perspective**

The ERS functions as a full-fledge interface between the student and the exam controller. While keeping the end user experience pretty simple and easily understood, it aims to achieve secure storage of the data collected from the students and convenient issue of hall ticket. It considerably reduces the time and effort of the students as well as the exam controller.

**2.2) Product Functions**

* The system provides a secure way of procuring personal information from the students and storing them in the database for further analysis of their eligibility to take up the examination.
* The system maintains user account which is protected by an alphanumeric password given by the respective user for securing the personal details.
* The system enables the user to choose from the available options to register for the examination of their choice and eligibility.
* The system prompts the user to provide scanned copies of their original identification documents to withhold the integrity of online registration.
* The system automatically generates and sends hall-tickets for the authorized candidates who can take up the examination by assigning an exam location which is in the chronological order of their city preference after the verification process.
* The system notifies the user through email upon status of their registration and approval of their application.

**2.3) User Characteristics**

* **Student**- The people who are interested in the enrolling process and fill out their personal details and choose the desired examination options. They upload the documents necessary for identity verification and are required to download the automated hall-tickets and produce the same when they are insisted to take up the examination.
* **Exam Controller**- The person who is entitled with the authority to add certain modules and constraints and decide who will take up the examination according to certain rules and regulations laid out by the institution conducting the examinations. The dispatch of hall-ticket can only be approved by him, upon meeting the criteria required such as successful payment of the examination fee and belongingness of the candidate to the institution that is responsible for the examination process.

**2.4) Constraints**

* It is necessary that the applicant has access to resources such as a computer and a stable internet connection to apply for the examinations.
* The user has to use his/her login credentials to proceed with further steps.
* Although the security of the user information is of high prominence, there is always an opportunity for intrusion in the web world which requires constant monitoring. Taking preventive steps to avoid the misuse of information is essential.
* The user has to be very careful in filling out the personal details because one submitted, it would be difficult to revert back the changes done.
* The scanned documents uploaded by the candidate have to be of certain minimum resolution to enable easy identification and a maximum limit to ensure sufficient storage.

**2.5) Assumptions and Dependencies**

* The students and the exam controller possess a basic knowledge of computers and how to go about operating the web interface according to their own requirements.
* The student is capable of scanning the documents and uploading the same within the given size so that it can be stored in the database and verified.
* The students should be able to carry out an online transaction using various payment methods such as UPI, debit card etc.
* The students should possess an user id for dispatching of hall-tickets and notifying them about the status of their exam registration application.

**3) Specific Requirements**

**3.1) Non-functional Requirements**

**3.1.1) Software Used**

* JDK - Java kit
* TomCat

**Coding Using**

* + - Java , Javascript, Java Servlet
    - Html and Css - Web designing

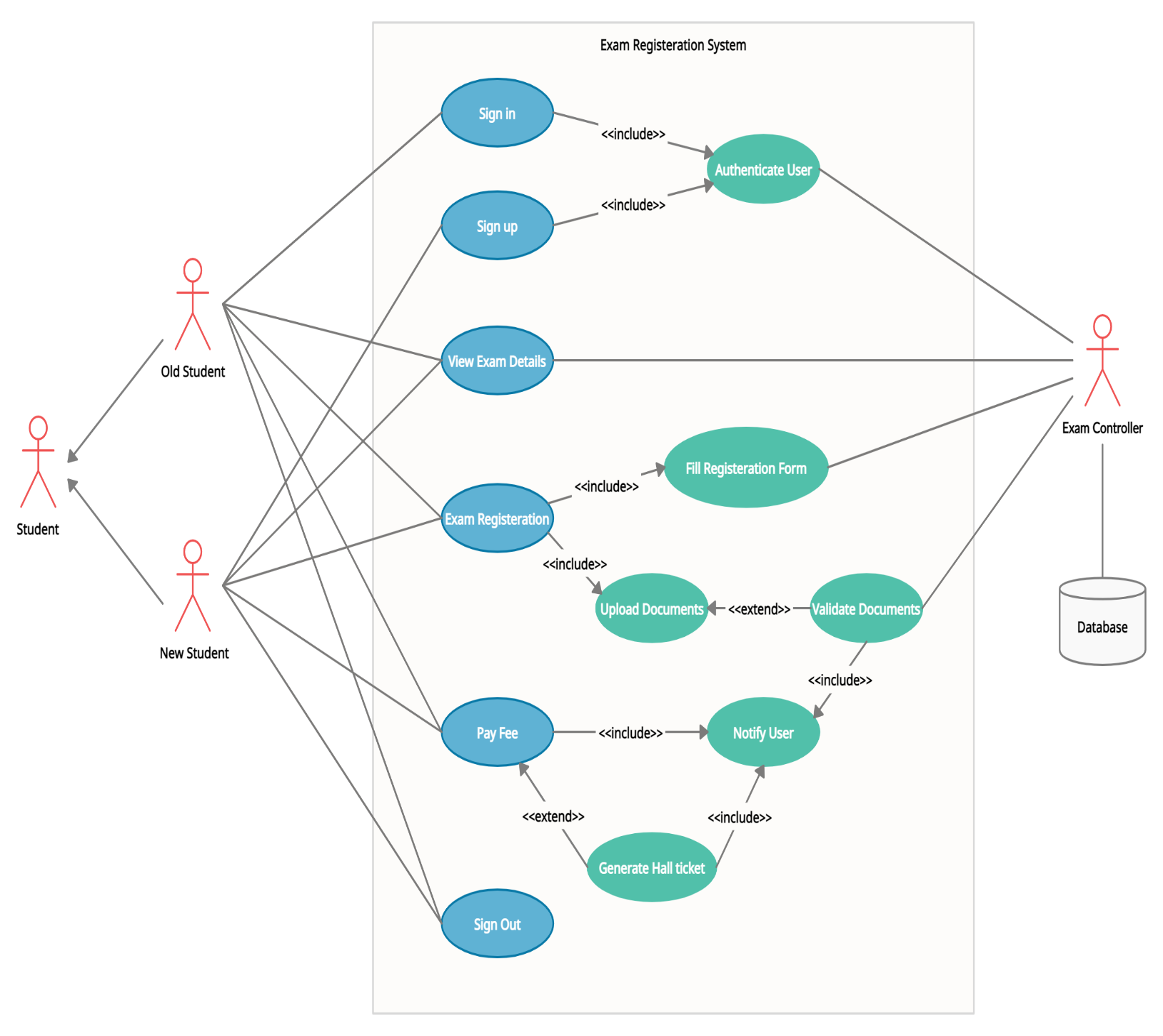
**3.1.2) Hardware Used**

The web server is directly connected to the client systems. The client systems can upload various documents within the size constraints and fill out their personal details to be stored in the database system of the serve

**USE CASE DIAGRAM**

**Notations:**

|  |  |
| --- | --- |
| **Symbol** | **Name** |
|  | Actor |
|  | Object |
|  | Association |
|  | Use case |
|  | Dependency |
|  | Generalization |
|  | Includes |

**Actors Involved-**

* Student
* New Student
* Old Student
* Exam Controller
* System Database

**Use-cases in the Exam Registration System**

1. Sign-in
2. Sign-up
3. Authenticate User
4. View Exam Details
5. Exam Registration
6. Upload Documents
7. Fill Registration Form
8. Validate Documents
9. Pay Fee
10. Notify User
11. Generate Hall-ticket
12. Sign-out

**USE CASE NAME: Sign-in**

**Overview:** Sign-In use case is for the purpose of enabling the student to access their exam registration portal.

**Actors**: Student, System Database.

**Frequency of Use:** 1

**Preconditions:** The student should possess an account and their credentials should be a part of the system database. The student should have stable internet connection.

**Main Success Scenario (or Basic Flow):**

1. Student opens the exam registration web server.
2. Student enters his username and password.
3. The student’s credentials match with the system database and he is able to sign in to his/her account.
4. The **Authenticate User use-case** is responsible for matching the credentials and authorizing access to the student.

**Postconditions**: The student will be redirected to the exam registration portal.

**Includes: Authenticate user**

**Extensions:** If the student is new to the web server, he or she has to go through the process of enrolling by entering their email id or user id and create a password for future access. **The Sign-Up use case** is responsible for taking the new user’s credentials and store the same on the system database.

**USE CASE NAME: Exam Registration**

**Overview:** The student has to fill the details and enrol for the examination by uploading valid documents that can be used as Identification proof.

**Actors**: Student, System Database, Exam Controller

**Frequency of Use:** 1

**Preconditions:** The student should possess an account and their information should be a part of the system database. The student should have valid identity proof and should be able to upload them while ensuring their storage is of the mentioned limit.

**Main Success Scenario (or Basic Flow):**

1. The student has to opt for the examinations he is going to appear for and fill the form according the same choices. The **Fill Registration Form use-case** is for the purpose of providing the appropriate form for the student.
2. The student has to enter their top three location preferences for taking up the examination.
3. The student has to scan and upload identification proof and photograph in the portal given for hall-ticket generation. The Upload Documents use-case interface is for the purpose of accepting valid and suitable-size documents and storing them in the system database for the exam controller to verify them according to the student’s identity.
4. Upon successful procurement of the student details, ask for confirmation by displaying their choices.
5. Notify the student upon successfully uploading the documents as well as the exam controller about the registration from the student.

**Postconditions:** The student will be redirected to the payment portal and will be intimated to initiate the payment process for the examination.

**Includes: Fill Registration Form, Upload Documents.**

**Extends:** Upload Documents extends Verify Documents through which the examination controller can approve or reject the application of the candidate.

**USE CASE NAME: View Exam Details**

**Overview:** The student has to choose from the given choices of examinations as per his credit requirements.

**Actors**: Student, System Database, Exam Controller

**Frequency of Use:** 1

**Preconditions:** The student should possess an account and their information should be a part of the system database. The student should choose the course of his domain of interest.

**Main Success Scenario (or Basic Flow):**

1. The system displays the courses or the examinations available for registration at the current time.
2. The course details and the syllabus for the examination along with the question paper pattern will be displayed.
3. The grade calculation information would be made transparent to the user.
4. The student can search if the examination can be taken up from a particular location.
5. Upon selecting a course for registering, the system updates the selected examination subject to the student’s information database.

**Postconditions:** The student will be redirected to the exam registration form to fill the details and upload his documents further.

**Includes:** None

**Extends:** None

**USE CASE NAME: Pay Fee**

**Overview:** The student will be provided with an interface for accepting payments through UPI, debit card etc.

**Actors**: Student, System Database, Exam Controller

**Frequency of Use:** 1

**Preconditions:** The student should possess an account and their information should be a part of the system database. The student should have access to a bank account through which payment can be securely done.

**Main Success Scenario (or Basic Flow):**

1. The system displays the courses enrolled and the total fee applicable for the examination according to the price fixed by the Exam Controller.
2. After verifying the summary of the payment and its split up, the student shall continue to next interface provided with options for payment.
3. The student shall select one of the provided payment methods and give necessary pin, UPI id or card details.
4. Upon successfully receiving the payment confirmation, the student will be notified with a payment receipt. The use case Notify User shall fulfil this purpose.

**Postconditions:** The student will be redirected to the hall-ticket status page which will be updated when the generation of the hall-ticket completes.

**Includes: Notify User**

**Extends:** None

**USE CASE NAME: Generate Hall-Ticket**

**Overview:** The hall-ticket to be produced when the candidate appears for the examination is generated and is made available to print.

**Actors**: Student, System Database, Exam Controller

**Frequency of Use:** 1

**Preconditions:** The student should have registered and paid the examination fee for the courses he is authorized the hall-ticket for.

**Main Success Scenario (or Basic Flow):**

1. The system displays the status of the generation of the hall-ticket and will notify the user when it is available.
2. The user will be presented with the hall-ticket mentioning the examinations he is allowed to take up and the location preference.
3. The process of generation of the hall-ticket will take place by seeking information from the student database and filling the examination centres in a chronological order.
4. If the location preference for a student is exhausted, he/she will be advised to take up the examination as per the second preference.
5. The student’s photograph will be included in the hall-ticket for identity verification purposes.

**Postconditions:** The student will be given an option to download the hall-ticket or print the hall-ticket accordingly.

**Includes:** Notify User

**Extends:** Pay Fee

**USE CASE NAME: Sign-out**

**Overview:** The student can save any changes done in the exam registration site and can log out using the sign-out use case.

**Actors**: Student, System Database.

**Frequency of Use:** 1

**Preconditions:** The student should possess an account in the exam registration website.

**Main Success Scenario (or Basic Flow):**

1. The system displays the sign-out option along with the account details.
2. The student can opt to sign-out if he/she has finished with the necessary procedures.
3. The student will be prompted with a message asking for confirmation of log out.
4. If yes, the account will be logged out and will require signing-in again to access the candidate’s examination portfolio.

**Postconditions:** The student will be redirected to the exam registration web portal with sign-in and sign-up options.

**Includes:** None

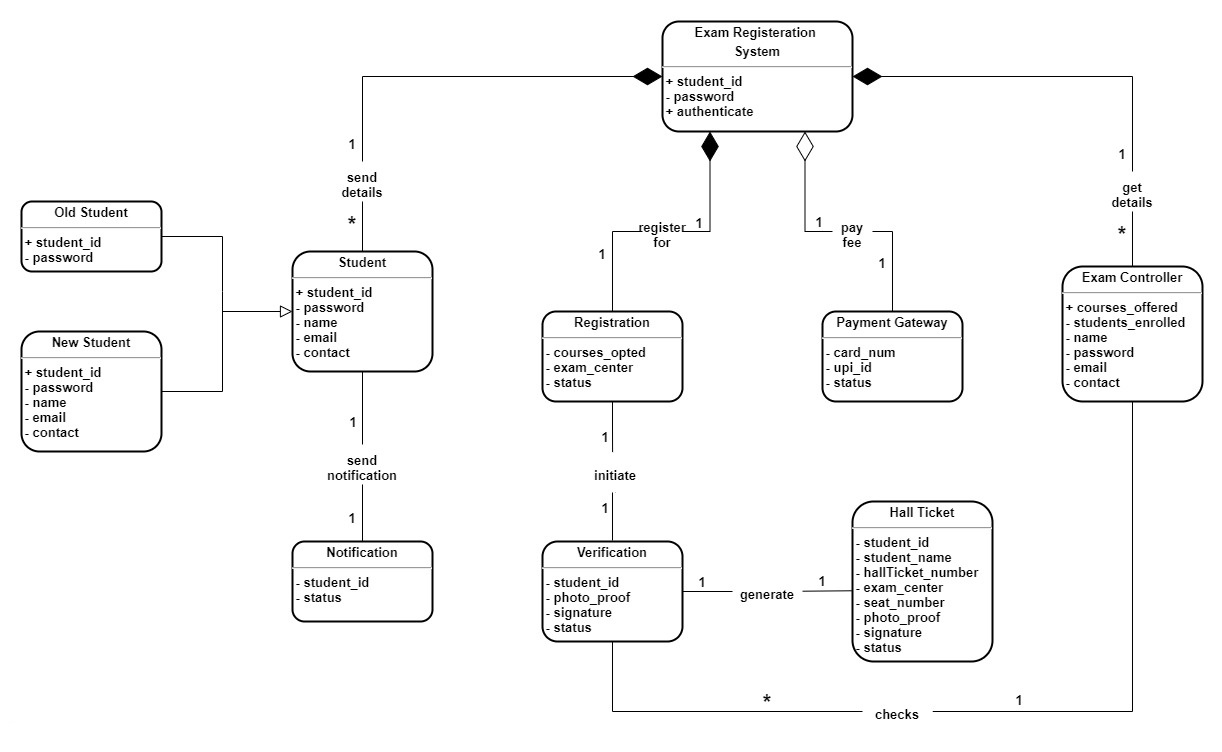
**Extends:** None

Conceptual Class and Domain Model

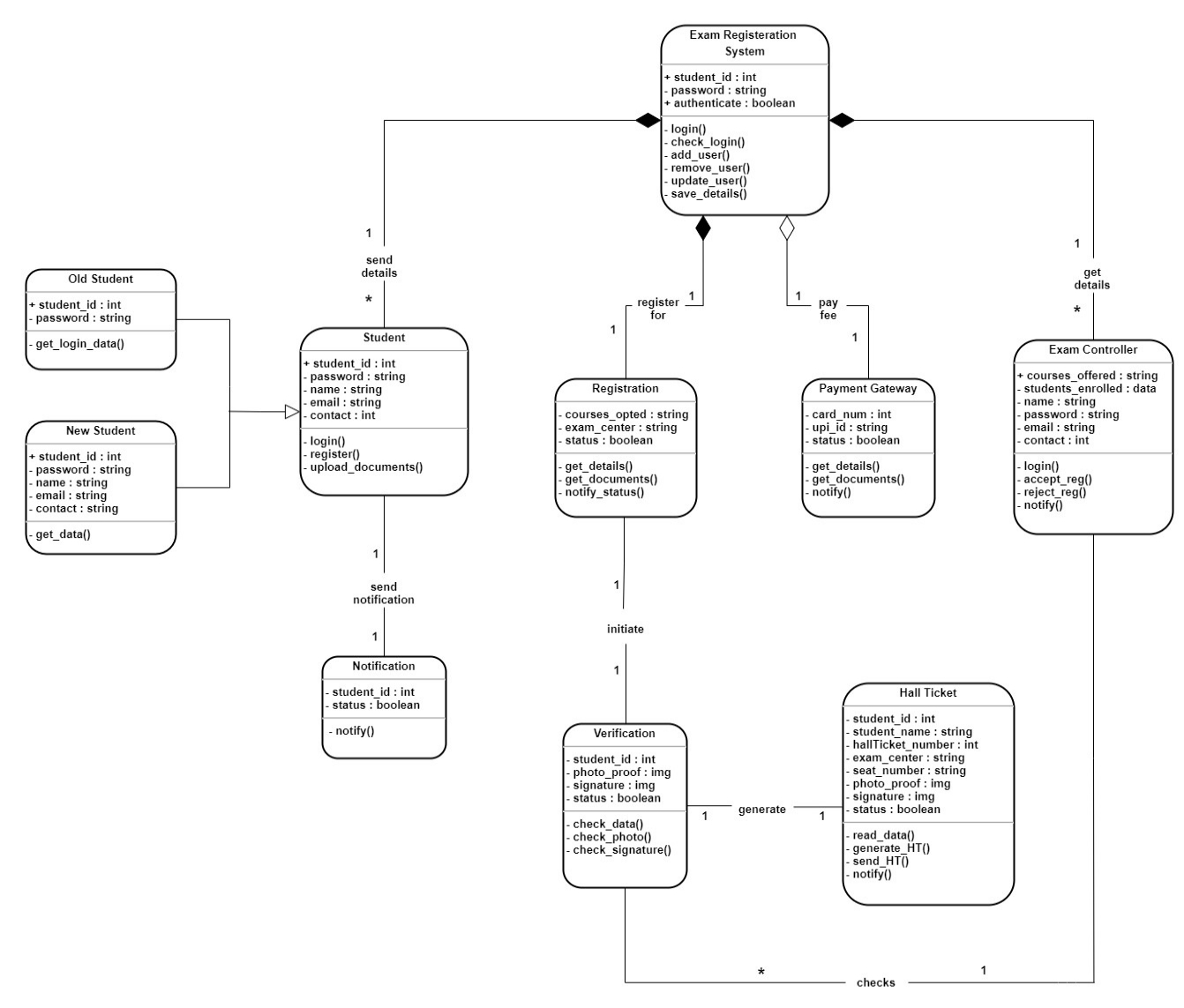
**UML Notations of Domain Model and Class Diagram:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NAME** | **NOTATION** | **DESCRIPTION** |
| 1 | Class | Class name | Class is an entity of class diagram.  It describes a group of objects with same properties and behavior. |
| 2 | Object | Object name: Class  name:Clas | An object is an instance of class |
| 3 | Generalization | Class2  Class1 | Generalization organizes classes by their super-class and sub-class relationship. |
| 4 | Aggregation | Class2  Class1 | Aggregation -one class is a part of another class |
| 5 | Composition | Class2  Class1 | Composition is a  Form of aggregation.  Composition implies ownership of the parts by whole |
| 6 | Abstract class | <<abstract>> | Abstract class has no direct instances |
| 7 | Association | Class2  Class1 | An association is a description of a  Link (conceptual connection among objects) |
| 8 | Multiplicity | Class1  \*  \*  Class2 | Multiplicity specifies the number of instances of one class that may relate to a single instance of an associated class |

**Domain model**



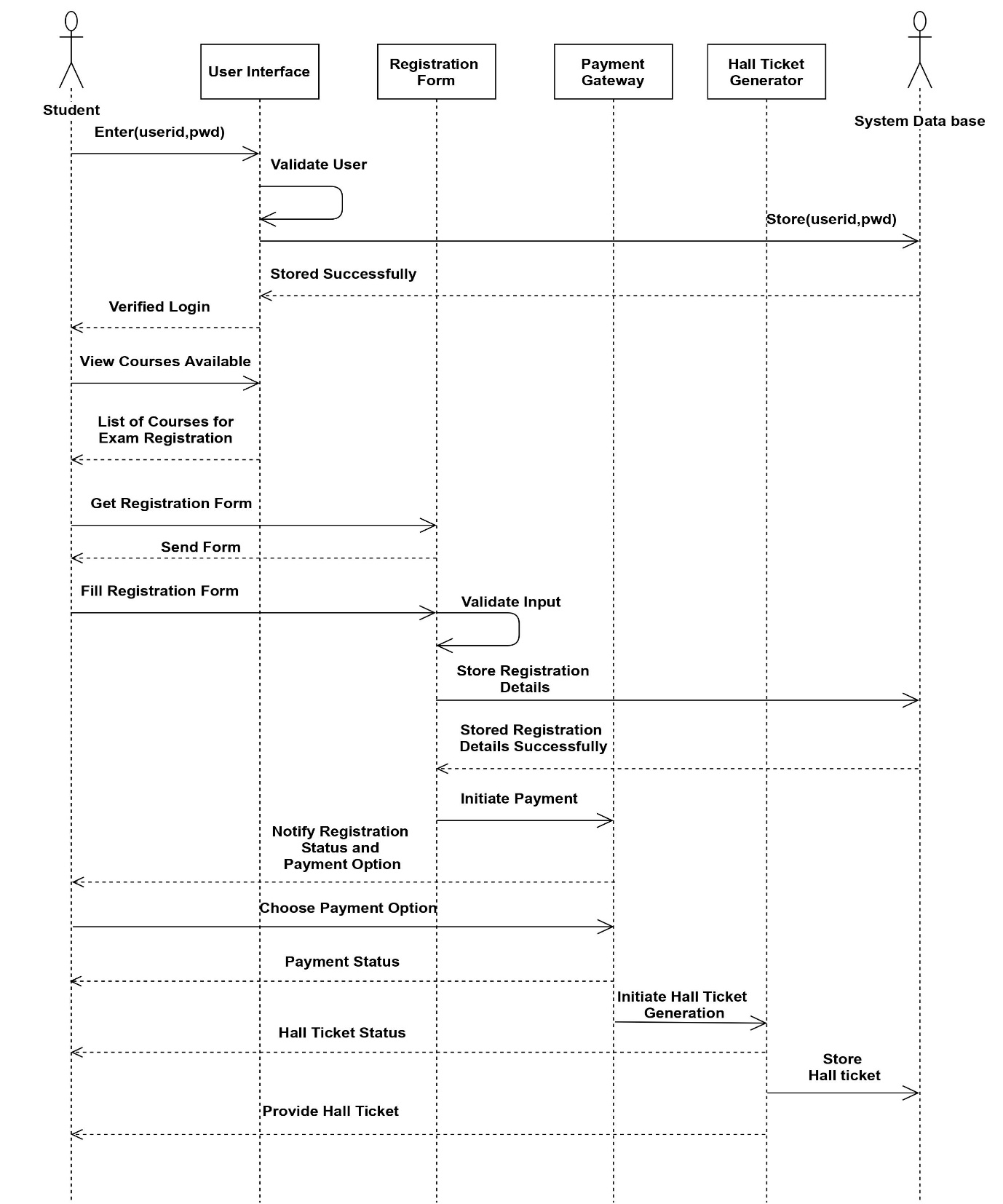
**Class Diagram**



Interaction Diagram

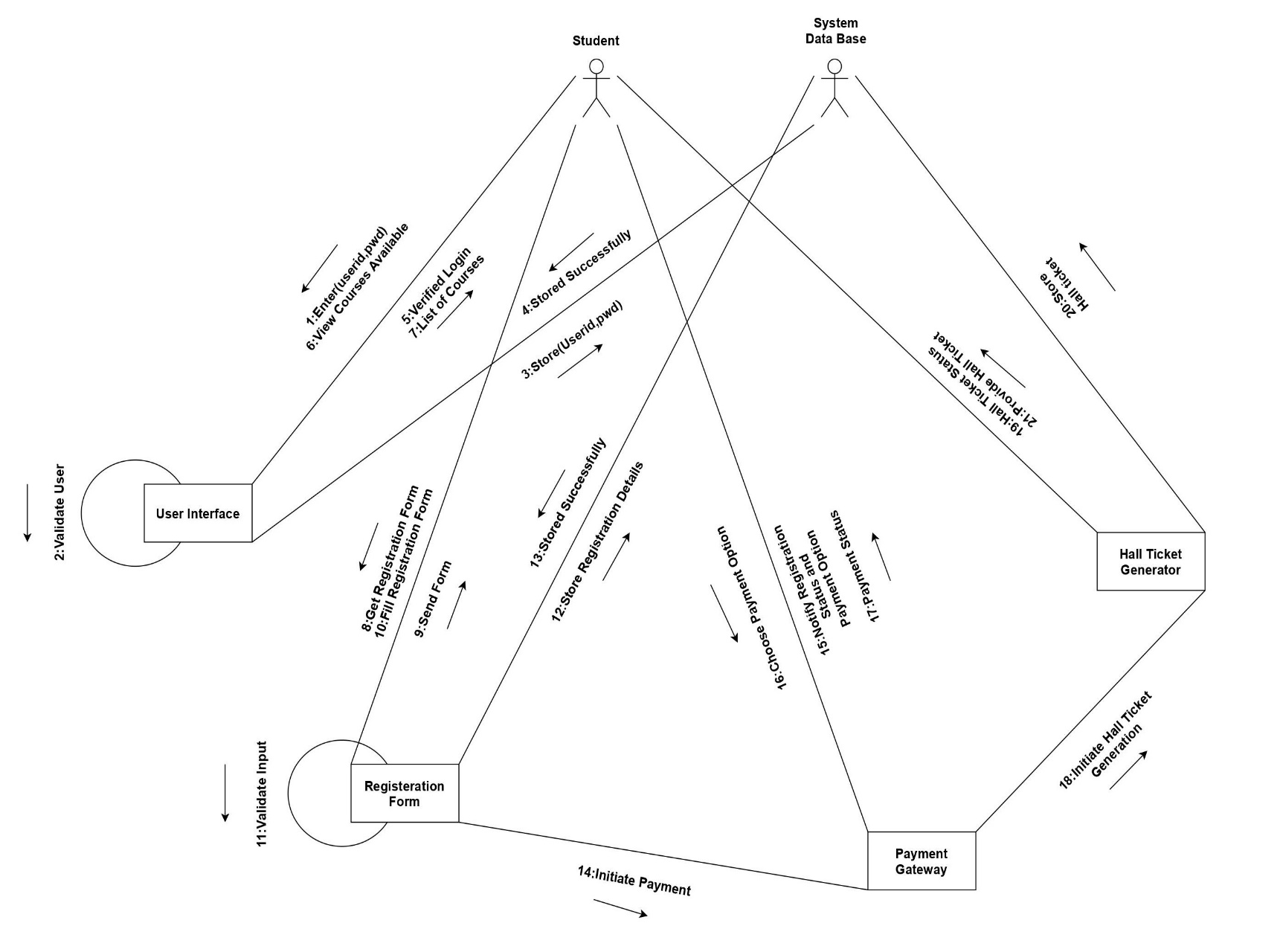
**Sequence Diagram :**

A sequence diagram simply depicts interaction between objects in a sequential order .Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.



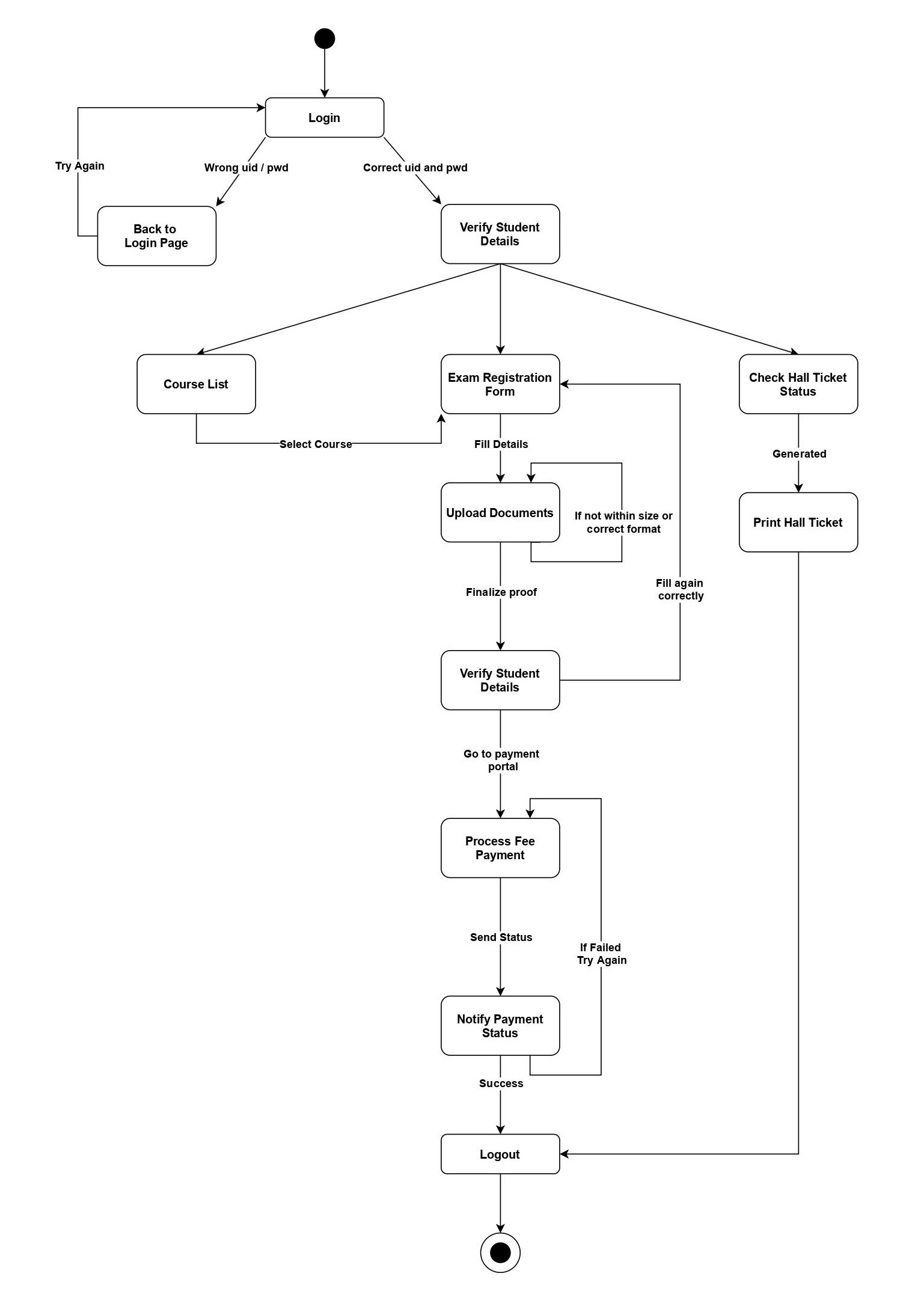
**Collaboration Diagram:**

A Collaboration diagram is easily represented by modeling objects in a system and representing the associations between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of these arrows.

**Collaboration Diagram**

**State Chart Diagram :**

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transition. A state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli.

State Chart Diagram

Activity, Package, Component and Deployment Diagram

Activity diagram Notations

|  |  |
| --- | --- |
| S.NO | Notations |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

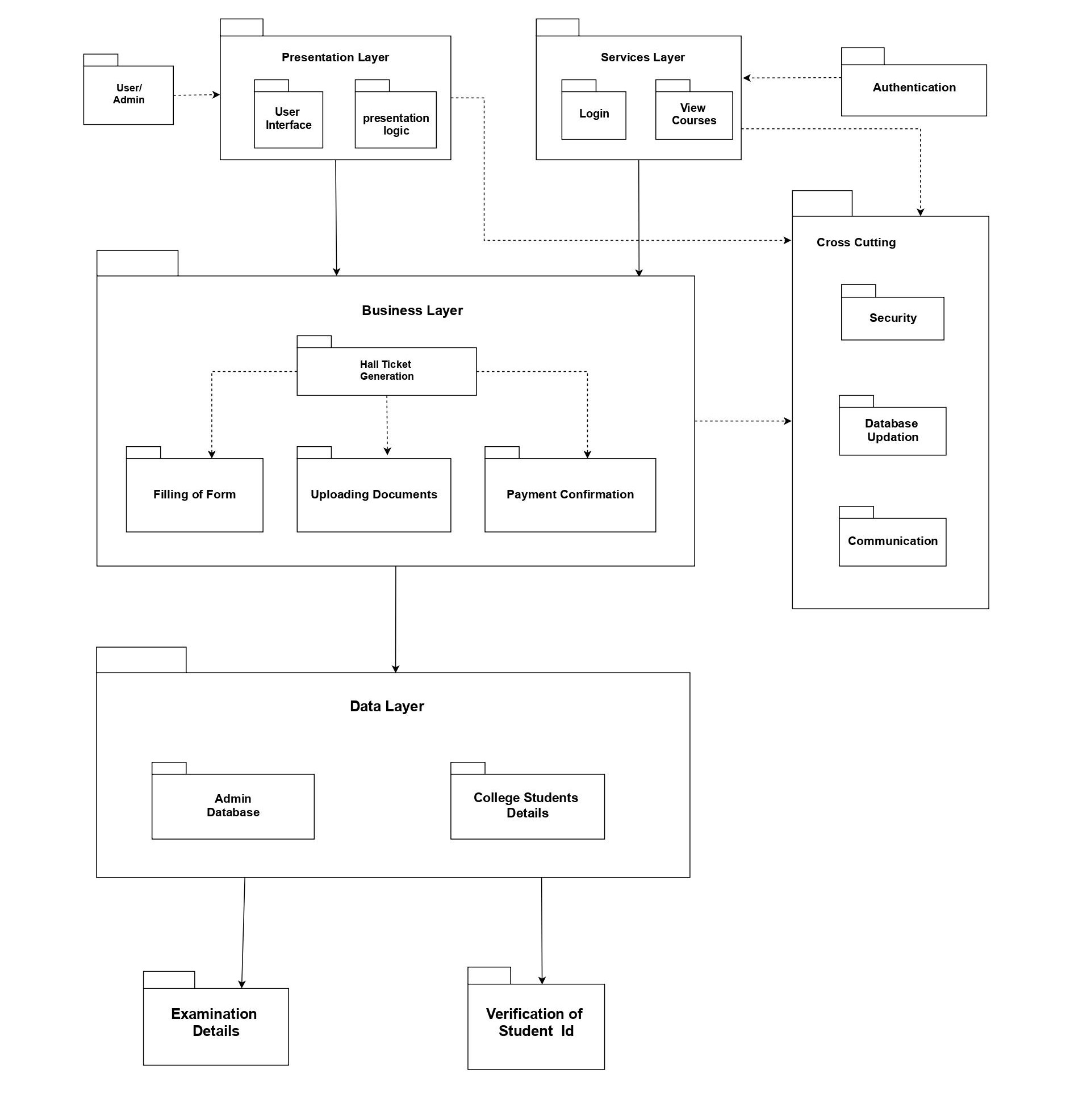
**Activity Diagram**



**PACKAGE DIAGRAM :**

A package is a grouping of model elements which means that a package can contain model elements of different kinds, including other packages to create hierarchies. A package defines a namespace for its contents using for various purposes.

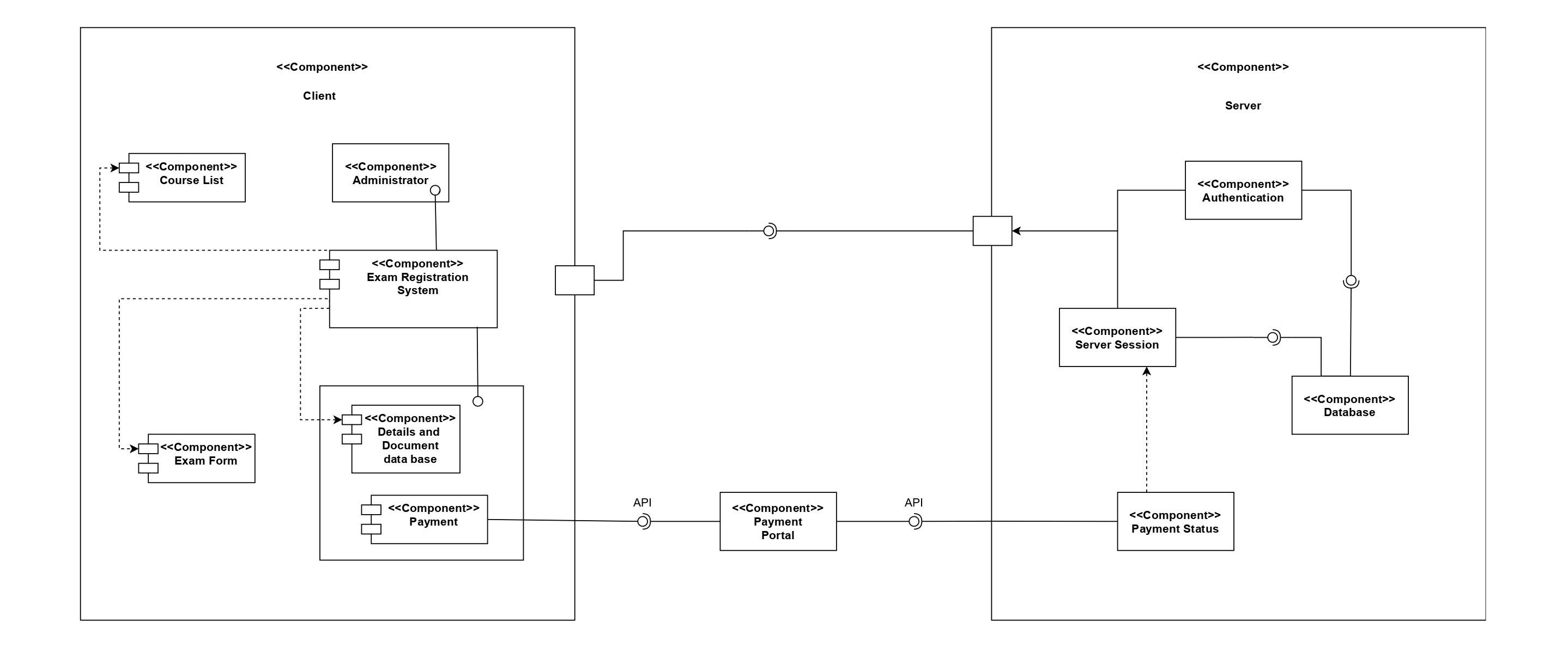
**Package Diagram**



**Component Diagram :**

It describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. In the first version of UML, components included in these diagrams were physical: documents, database table, files, and executables, all physical elements with a location.

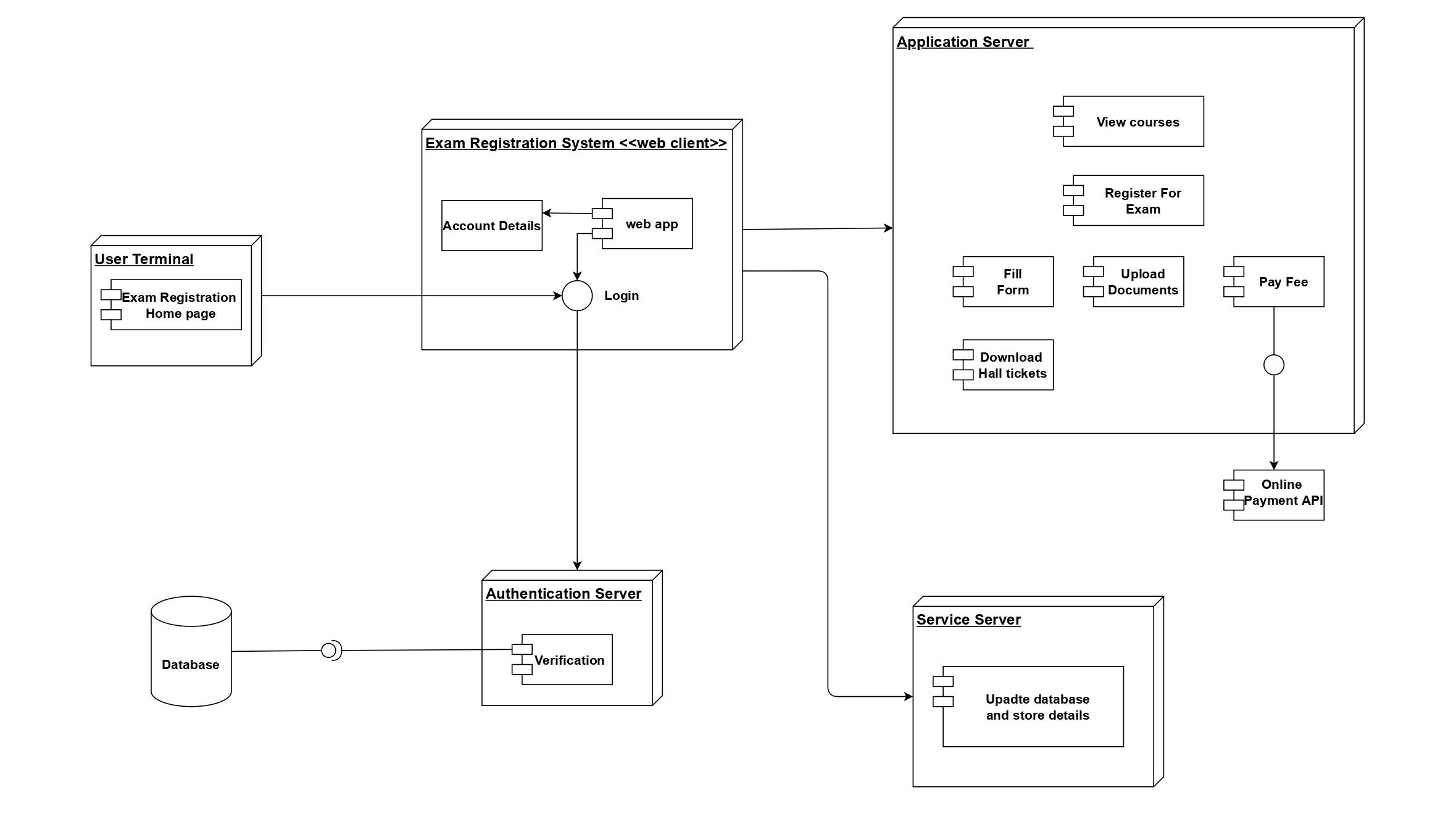
|  |  |
| --- | --- |
| **S.NO** | **Notations** |
| **1** | Component symbol |
| **2** | Interface symbol |
| **3** | Dependency symbol |
| **4** |  |

**Component Diagram**

**DEPLOYMENT DIAGRAM :**

It is used to visualize the static aspect of these physical nodes and their relationships and to specify their details for construction. A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of an architecture.

**Deployment Diagram**

****