University Management System

***Project Description:***

A system that will be used by university’s staff and students to deal with student’s information, course registration, grades, fees, courses offered by each professor. It will also be used by human resource personnel to keep track of professor’s information and calculate their salaries according to number of courses offered. All of the data will be saved on a SQL database and fetched using relational tables.

***Team members:***

*ismail yasser ismail abdo 21100783 <leader>*

*ali yasser ali ali 21100801*

*Amira Ali Mohamed Hafez 21100789*

*Reham Atif hussein 20100333*

*Aya Adel saied gomaa 21100786*

***Initial Customer Requirements:***

1.Students:

* Login with a username and password.
* Register courses according to the allowed CRD hours.
* View Registered courses for the semester and view the coursework marks.
* View Final grades of each course after the course if finished, and view the term’s GPA as well as the total GPA.
* View the total fees for the semester according to the registered courses and whether its paid or not.

2.Professors:

* Login with a username and password.
* Select courses to offer.
* View each course offered.
* View all students enrolled in each course.
* Assign marks for each student, when the course is finished and all marks are submitted the system automatically will calculate the student’s grade.

3.Human Resources Personnel:

* Login with a username and password.
* View all professors and their information.
* View each professor’s salary according to number of courses offered.

4.Administration:

* Add students, professors and H.R to the system.
* Delete a student’s registration, add a student’s registration.

***Initial Test Requirements:***

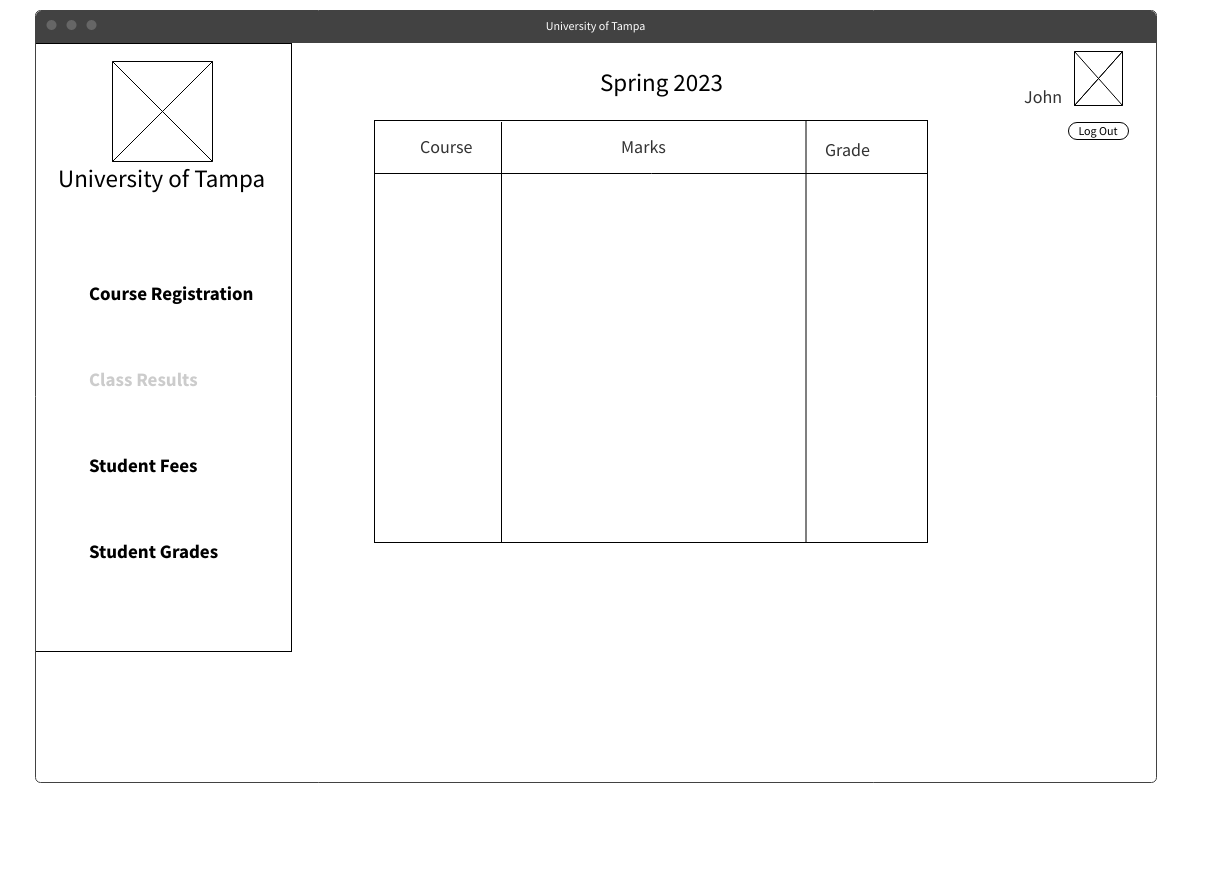
* Validate login credentials and sign in to the correct account.
* Test registering more than the allowed credit hours as a student.
* Validate that each student can only access their data/grades/marks.
* Validate that the student fees are calculated correctly.
* Validate that each professor can view all the course they registered nothing more or less.
* Validate that all the students enrolled in a class are viewable by the professor.
* Validate that when a professor submits the marks/grades of students, they get updated on the system.
* Test the search for H.R accounts
* Validate that the salary of the professors is calculated correctly.
* Validate that when an admin adds an account, its updated on the system and can be logged into by the user.

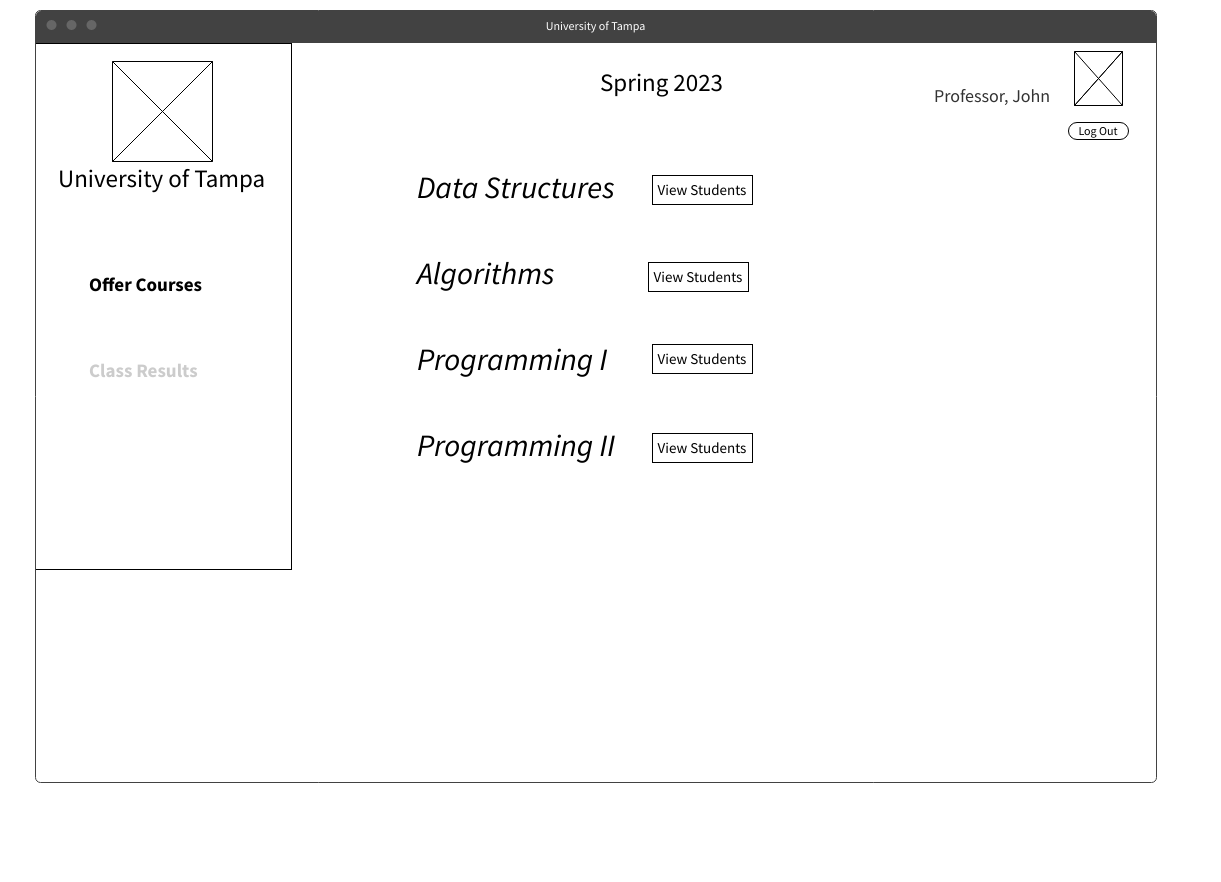
***Traceability Matrix:***

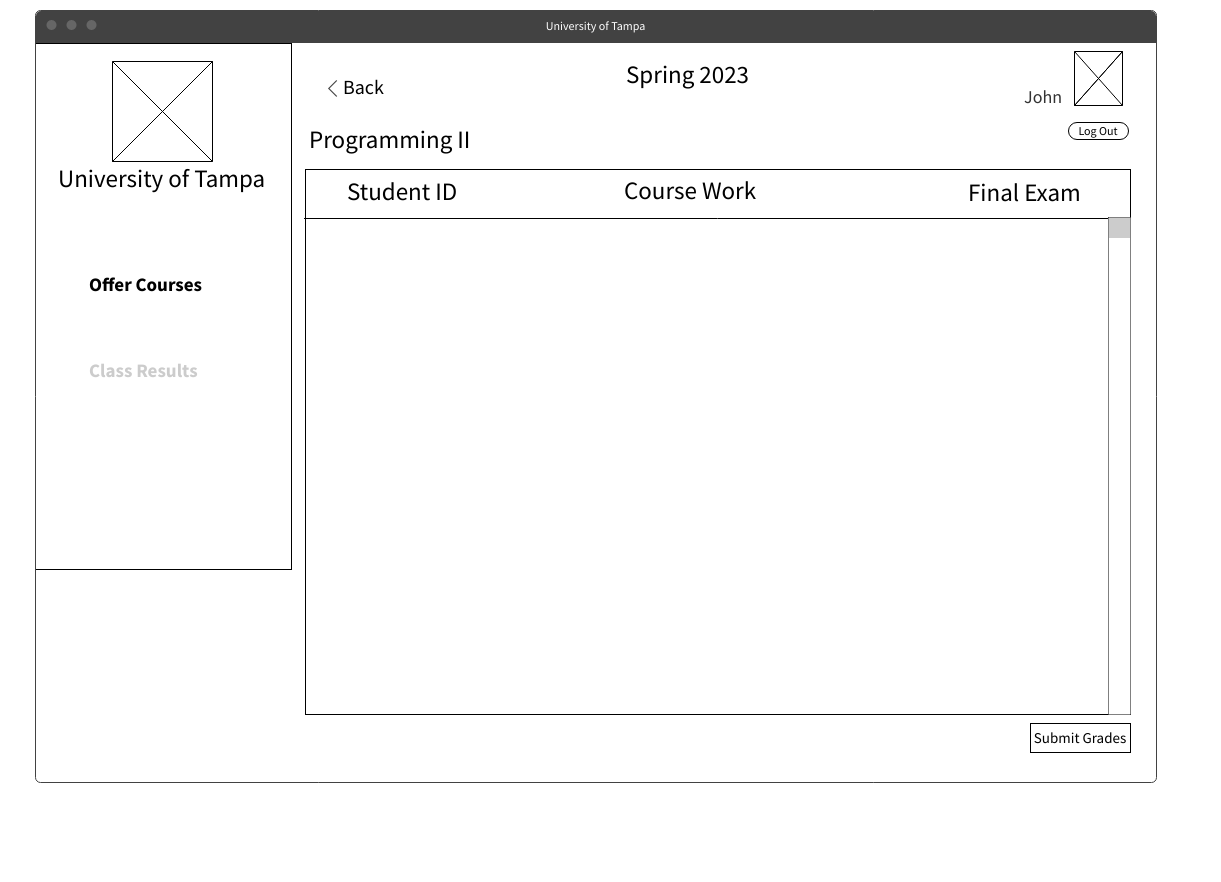
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req. no. | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 3.1 | 3.2 | 3.3 | 4.1 | 4.2 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

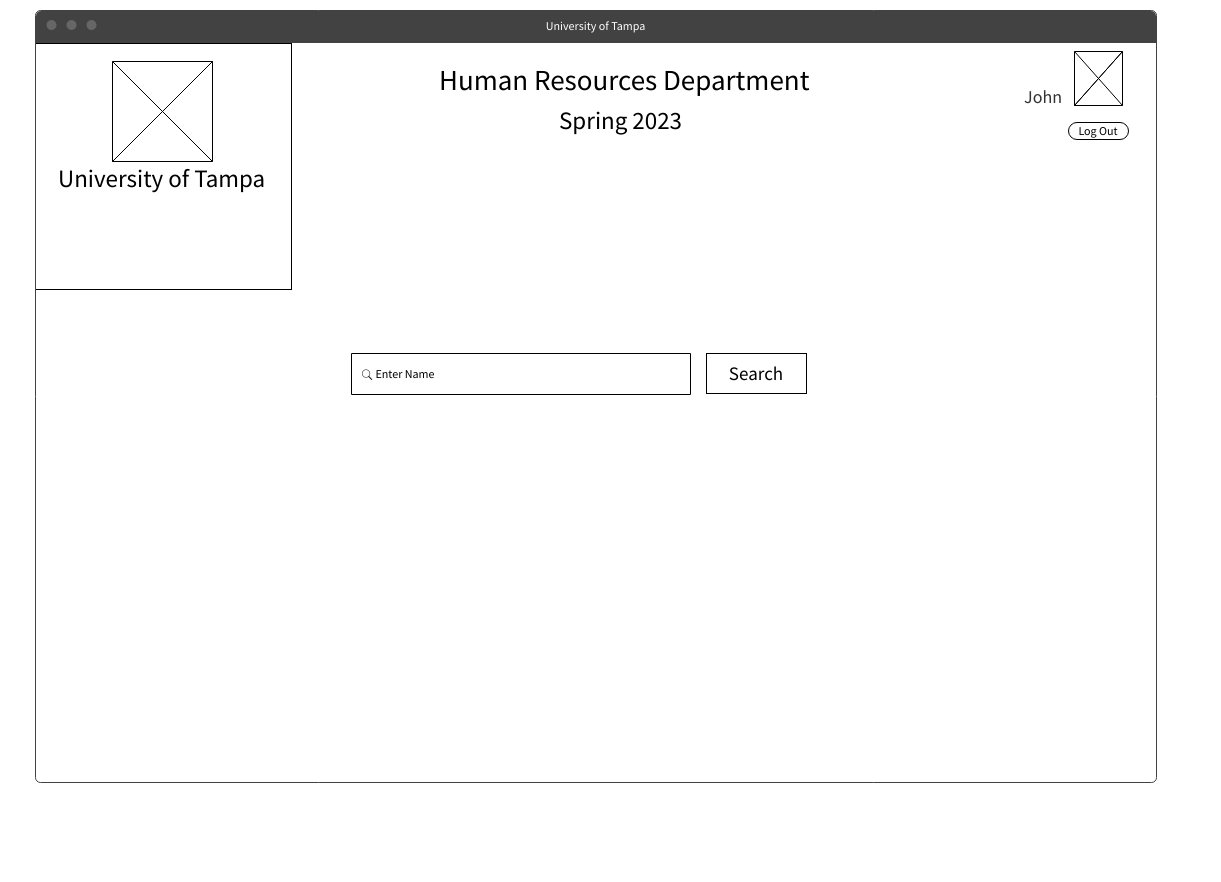
***Initial Subsystems:***

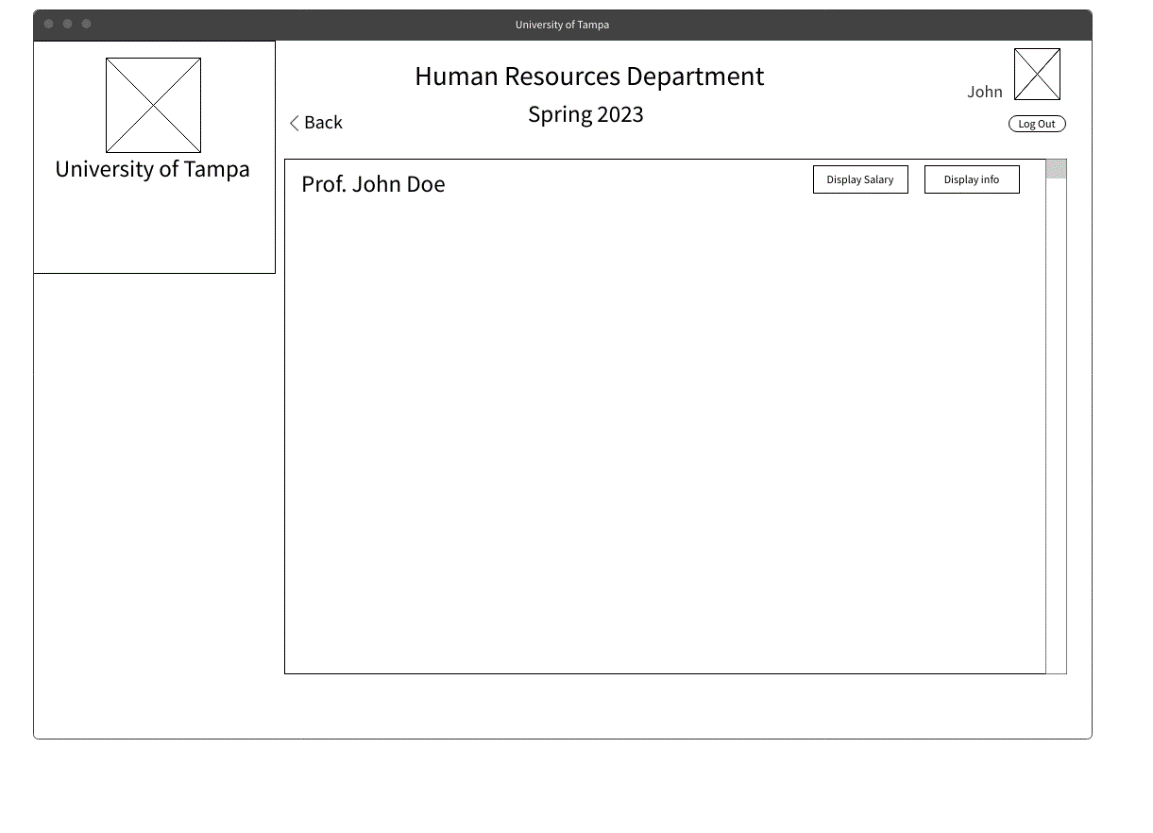
|  |  |  |
| --- | --- | --- |
| Subsystem Name | Subsystem Function | Subsystem Interface |
| Course Registration | Add the selected courses by the user (student) to the database only if the number of credit hours is within the limit that the user can register | RegisterCourseInterface:   * void register(List[courses],int crdHrs) |
| Calculate Student GPA | Calculate the GPA of the student both semester and total. | CalculateStudentGPAInterface:   * float calcGpa(List[courses]) |
| Calculate Student Fees | Calculate the student’s fees for the semester | CalculateStudentFeesInterface:   * float calcFees(int crdHrs,float fees) |
| Calculate Professor’s Salary | Calculate the salary for each professor | CalculateProfessorsSalacryInterface:   * float calcSalary(int numCourses, float pricePerCourse) |
| Transportation management | scheduling transportation routes and pickups/drop-offs for students and staff, tracking vehicle maintenance and fuel usage, and ensuring compliance with transportation regulations and policies | Transportationmanagement interface:   * void transportation(String name ,   Int id , String address , int carid) |
| Reporting system | This subsystem should provide a mechanism for reporting incidents of misconduct or violation of policies. The reporting system should be accessible to all stakeholders, including students, faculty, staff, and administrators. The subsystem should also ensure that reports are treated in a confidential, respectful, and timely manner. | Reportingsystem interface :   * Void report(String Name ,   Int id , String info) |
| Payroll management | responsible for the salaries of dr.s, teaching assistants, and all employees within the university ,as well as increases , discounts, and bonuses, if available | Payrollmanagement interface:   * Float pay(Array[Staff],   Array[students],float fees) |
| Clinic system | responsible for the medical examination of students in applying to the university and emergency cases for students and anyone inside the university during the university days | Clinicsystem interface :   * Void clinic(String name ,   Int id , String faculity) |
| Activity management | responsible for student activities ,organizing parties ,events and other recreational activities at the university | Activitymanagement interface:   * Void activity(int noofstudent,   Float price ) |
| Leave and attendance management | responsible for vacations for students, teaching staff and workers at the university ,and recording attendance daily | Leaveandattendance management interface :   * Void attendance(Sting name,   Int ID) |
| Student management | This subsystem would allow administrators to manage student records, including personal information, enrollment status, academic progress, and more. It could also include tools for tracking student attendance, issuing grades, and generating reports. | Studentmanagement interface :   * Void student(String info ,   Float gpa) |
| Course Management System | This subsystem would allow instructors to create and manage course materials, including syllabi, assignments, and exams. It could also include features to track student progress, grade assignments, and provide feedback. | CourseManagement interface :   * Void course(int crdhours ,   String coursename ) |
| Library Management System | This subsystem would allow librarians to manage the university's library resources, including books, journals, and other materials. It could include features to track borrowing and returns, manage subscriptions, and provide access to online | LibraryManagement interface :   * Void library (String membership ,   String name , inti D) |

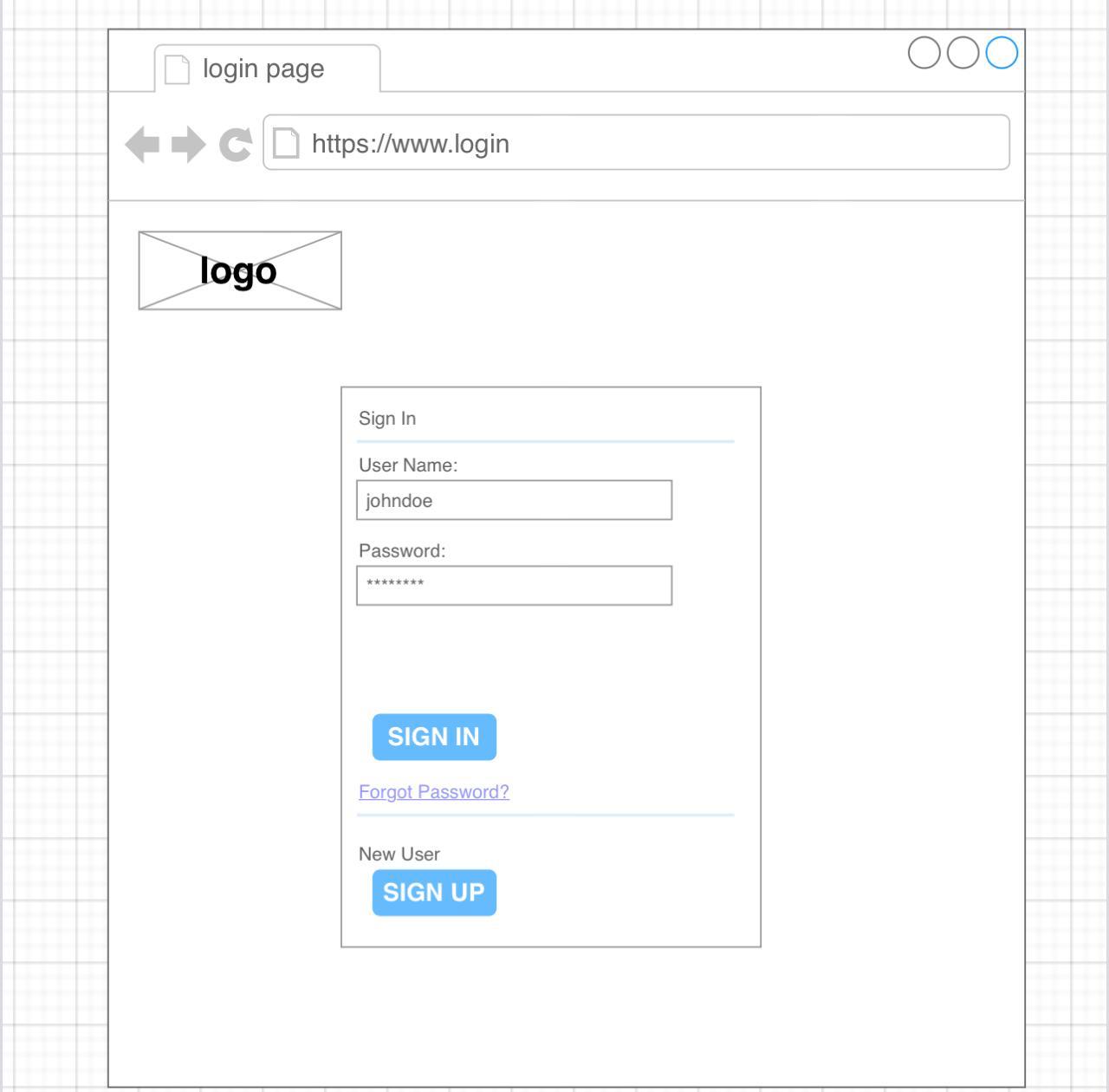
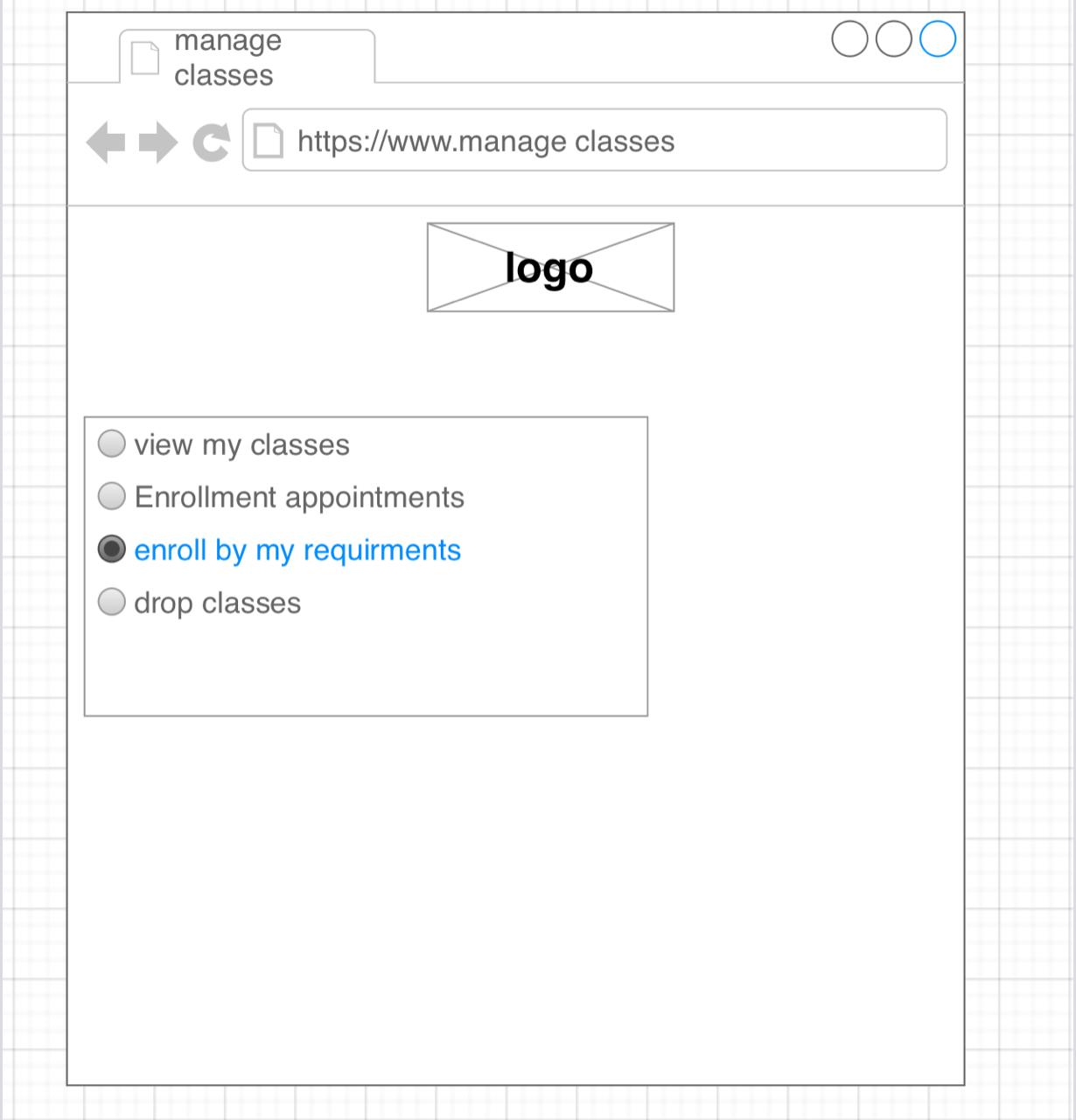
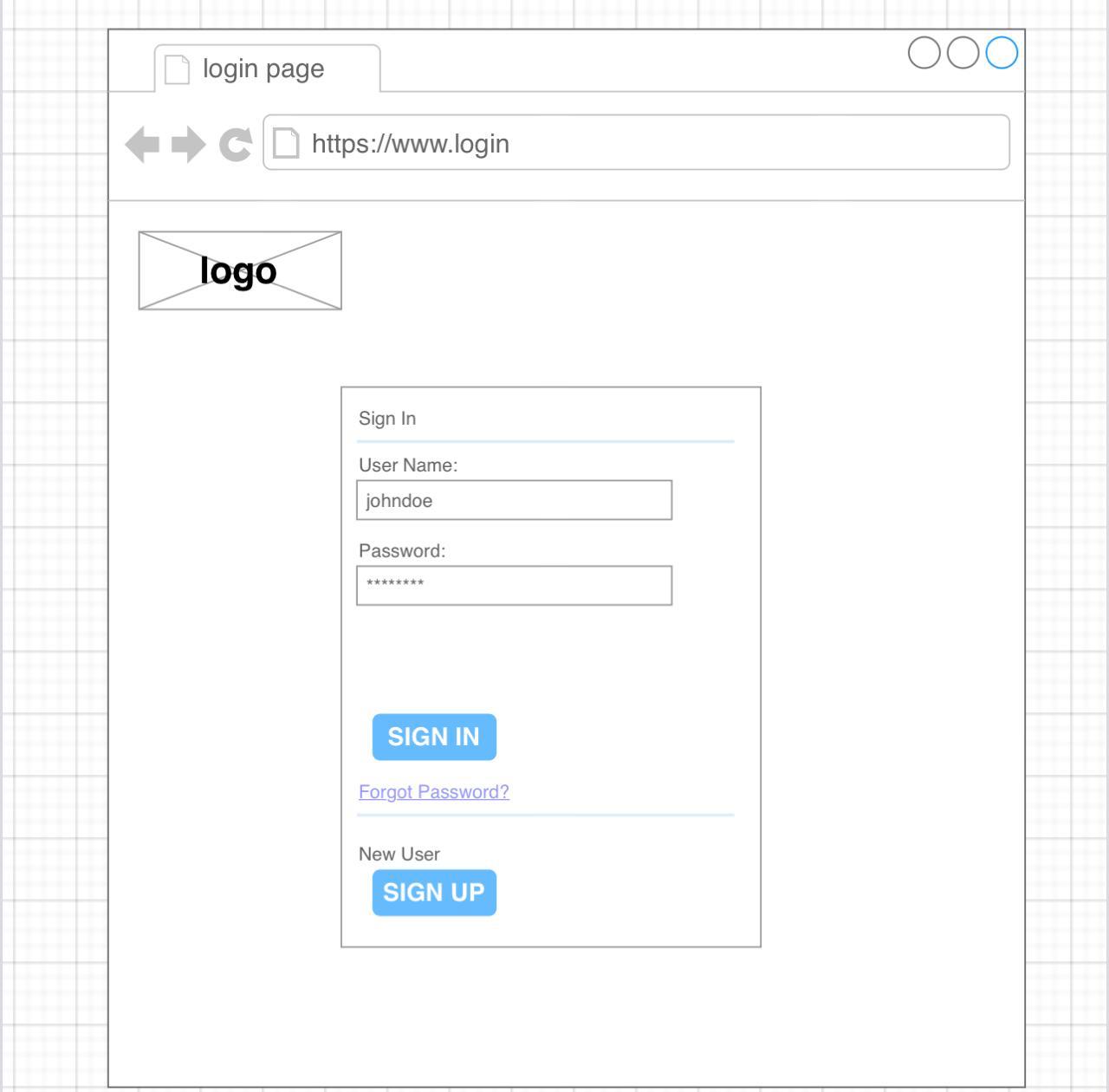
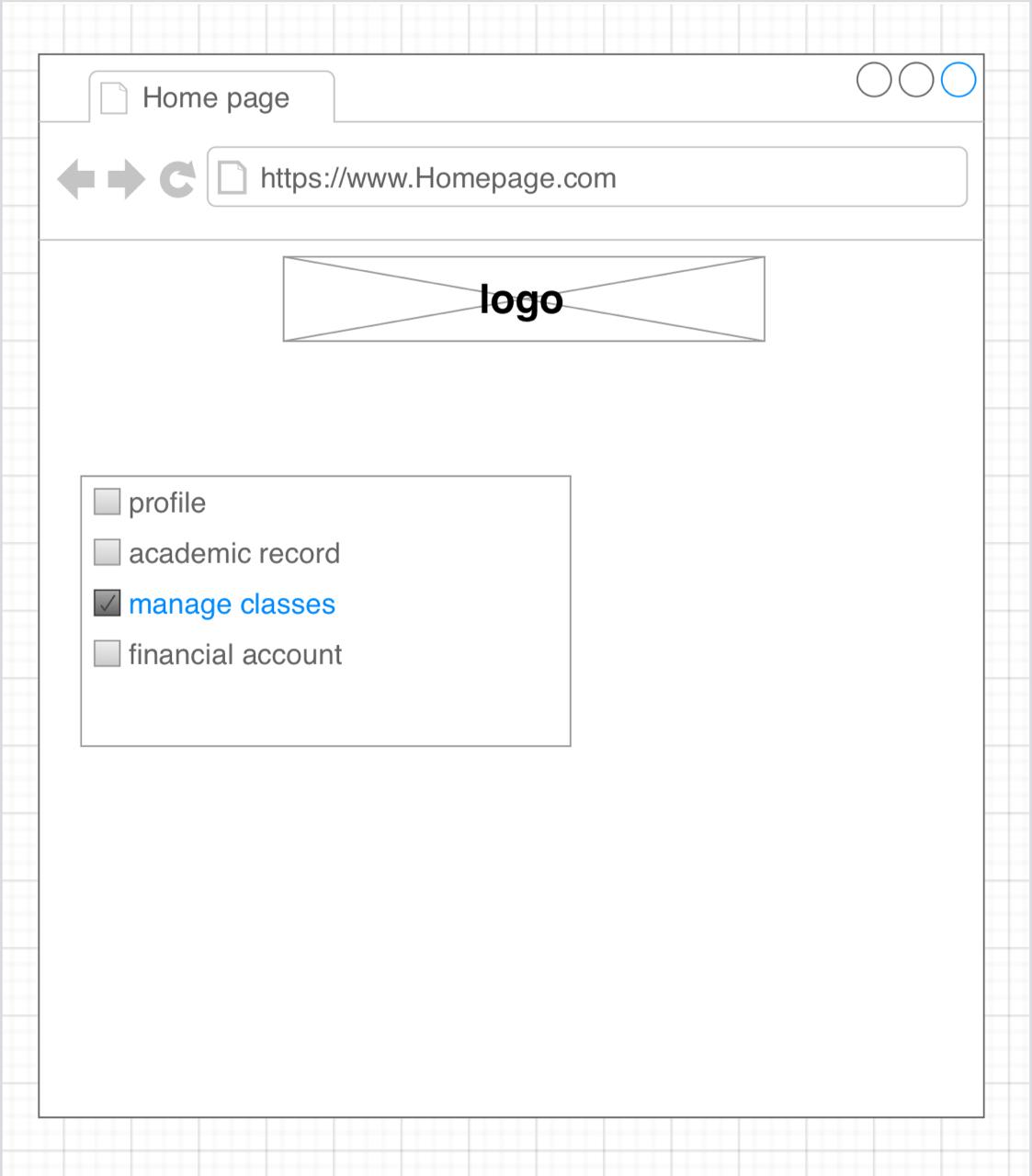
***GUI Sketch:***











***User Manual:***

Student

* Sign up with username and password after selecting the option ‘student’.
* When the university announces the time for course registration you can access the course registration tab and select the courses you want to take this semester.
* Your home screen displays the courses you are enrolled in this semester and you can track your course work marks.
* Select Student Fees tab to display your fees for this semester and whether its paid or not.
* Select Student Grades tab to display all your grades across semesters and your GPA.

Professor

* Sign up with username and password after selecting the option ‘Professor’.
* Your home screen is the courses you offer for this semester, click ‘view students’ on any of these courses and it will take you to a page that displays all students that are enrolled in this class.
* You can assign marks to each individual student for coursework and final exam.
* After all the final exam marks have been assigned, you can submit grades to the system to be viewed by students.
* You can select the tab ‘offer courses’ at the start of the semester to select which courses you will be offering.

H.R Employee

* Sign up with username and password after selecting the option ‘H.R’.
* In your home screen you can search the professors in the database by name.
* You can view details of each professor and salary for this semester to initiate payments.

Login

* Select llink in this link thereis login page to make sign in or sign up

Functional and nonfunctional :

Functional :

**Login into the Portal.DESC** : The user can login into the system by providing its login id and password to the portal which are unique and provided by the college.RAT:To login into the portal and take advantage of the services provided by it.

**Change passwordDESC** : Change the password after logging into the system.The user will be asked a securityquestion if he/she forgets his/her password. In case, he/she is unable to do so, the user has toformally apply to the admin for the change of his/her password providing the necessary proof.RAT:For security purposes, the user can choose his/her password

**View/change proﬁle details.DESC** : The user can view or change some of his/her personal details like email id, contact detailsand address details. The proﬁle will contain name, age,permanent address, parent’s name, theiraddress, their contact details, branch, year, semester, room alloted, hostel name and no. etc.RAT:To update his/her proﬁle and know his or her status.

**HelpDESC:** Can get help through the help option which gives information about the diﬀerent fea-tures provided by the system. A copy of user manual will also be provided in this section.RAT:In case of any confusion, the user can solve the problem easily and better understand thefunctionality of the system

**TimetablesDESC:** The user can view timetable for classes, mess and buses on the front page of the portalafter logging into the system.RAT : By geting this information, user can schedule his routine and better utilize his/her precioustime

**List of upcoming holidaysDESC**: The user can view the list on upcoming holidays on the front page of the portal afterlogging into the system.RAT:He/she could plan for vacations and other co-curricular activites

**AcademicsDESC**: This module on dashboard contains all the information about the academics and admin-istration.RAT : You can get all the information about academics after getting inside this module.

**Request for certiﬁcatesDESC**: The user can request for diﬀerent types of certiﬁcates After ﬁlling the given format , the request willbe send to the administrative department for veriﬁcation of request. After the veriﬁcation, theuser can get his/her certiﬁcate

**FacultyDESC**: This module provides the information about the faculties. This includes name, educationdetails, areas of interest and expertise, email id and contact details(optional).RAT:You can get all the information about faculties in one place after getting inside this mod-ule

**Assignments and reading referencesDESC**: The user can download the reading and writing assignments uploaded by the faculty fortheir course.RAT:The user will have a remote access to the assignments

**Course EvaluationDESC**: At the end of the semester, a new page will be available for evaluation of a particularcourse for the students. The user can give evalution for a particular course by just ﬁlling themultiple choice questions and submitting.RAT:This helps faculties to get a feedback for their course and improve their performance byintrospecting and also it helps the administration in better allocation of the faculty.

**Exam ScheduleDESC**: The user can view the exam schedule.RAT:By using this feature,the user can prepare and plan his eﬀorts in a better way

**Fee Receipt GenerationDESC**: Student can generate fee receipt by giving the bank details in the system authenticatedby the administrative head for the current and the previous semesters.RAT:This feature makes the process move convinient,fast and less cumbersome

**ResultDESC**: Student can access their results of past semester and generate a pdf ﬁle and print bylogging into the portal. The user can not view results of other students.RAT:The user can easily keep track of his/her academic performance

**Course RegistrationDESC**:Here, the user can register for the courses at the start of the semester. This page will beopen for a very short time interval. The user has to provide all the necessary details and selectthe electives for the semester.RAT:The user can easily register for the semester without cumbersome paperwork

**Exam ScheduleDESC**:The user can view the exam schedule. The user have the right to update/ modify theexam schedule by either automatically generating the timetable by providing the required detailsor manually ﬁlling the blanks.RAT:By using this feature,the user can prepares a perfect timetable without any conﬂicts.It isvery time eﬃcient

**Register new students.**

**Record the attendance of students.**

**Record the internal marks of students.**

**Record the feed details of students.**

**Register a new teacher/employee.**

**Register a new user for the system.**

**Record the salary details of employees.**

**Record the course details and subject information.**

**Record the scholarship details and information.**

**Generate various reports for all transactions in the system.**

**3.3.1 Performance RequirementsPerformance should not be an issue because all of our server queries involve small pieces ofdata.Changing screens will require very little computation and thus will occur very quickly.Serverupdates should only take a few seconds as long as the phone can maintain a steady signal.**

**3.3.2 ReliabilityMust maintain data integrity. Computer crashes and misuse should not aﬀect a user’s history**

**3.3.3 AvailabilityThe CMS Portal shall be available, up and running for 24\*7 throughout the year except due tothe routine maintenance activities**

**3.3.4 Security RequirementsAdministrator and Users with valid credentials will be able to log in to Portal.Administratorwill have access to the database structures at back-end.Administrator will have the rights formodiﬁcations as well as any Updation work for the datasets and website. Access to the varioussubsystems will be protected by a user log in screen that requires a user name and password.Tobe updated in future.**

**Non-functional Requirements:**

The system must be available 24/7, with minimal downtime for maintenance.

The system must be able to handle a large number of users and user activity without slowing down.

The system must be secure and protect user data and personal information.

The system must be easy to use and navigate, with intuitive interfaces and clear instructions.

The system must be compatible with a variety of devices and platforms, including desktop and mobile.

The system must be accessible to users with disabilities, including visually impaired users.

The system must be scalable, allowing for future growth and expansion.

The system must be able to backup and restore data to prevent data loss in the event of a system failure.

The system must comply with relevant laws and regulations, including privacy laws and copyright laws.

The system must be customizable, allowing users to personalize their experience and settings.

**Privacy:** The system should comply with privacy laws and regulations, and ensure that student and faculty data is kept confidential.

Data Backup and Recovery: The system should have a robust backup and recovery system to prevent data loss in case of a system failure or disaster.

Interoperability: The system should be able to integrate with other university systems, such as HR and financial systems, to ensure efficient data sharing.

Internationalization: The system should support multiple languages and currencies to cater to a diverse user base.

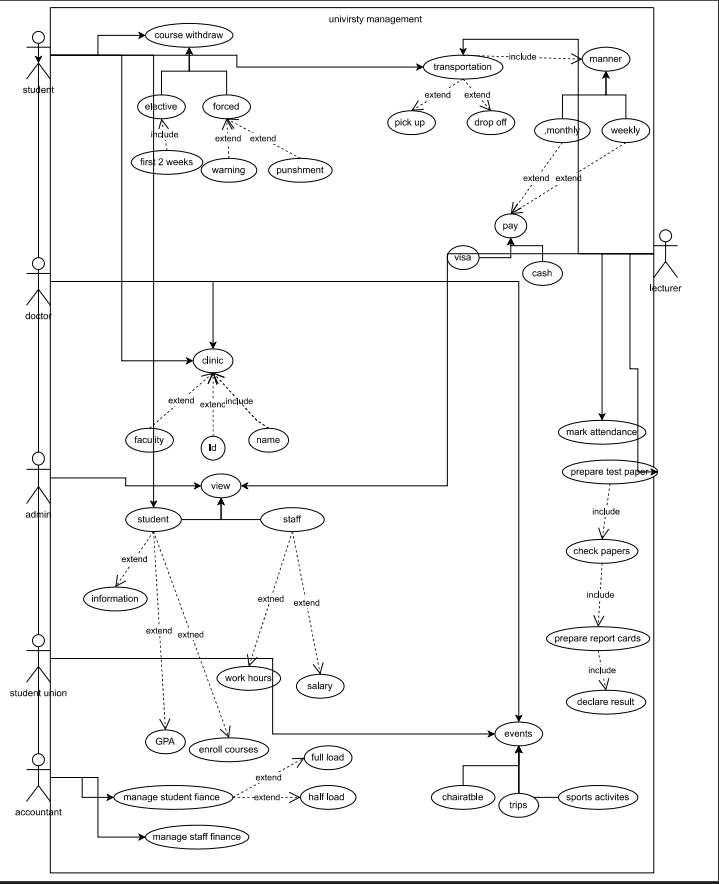
Response Time: The system should provide quick response times to user requests and queries to ensure user satisfaction.

User Training and Support: The system should provide user training and support to ensure users are able to use the system effectively.

Compliance: The system should comply with all relevant laws and regulations, including data protection and accessibility standards.

TEST REQURMINTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course Withdraw | pay | Transportation | Events | Pay and Manage  Student and Finance |
| Test Requirements:  1. The system should allow the student  to withdraw from a course by  accessing the course management  page.  2. The system should display a  confirmation dialog box to the student  before withdrawing from the course.  3. The system should prevent the  student from withdrawing from the  course if they have already completed  the course.  4. The system should prevent the  student from withdrawing from the  course if the withdrawal deadline has  passed. | 1. The system should provide  administrators with access to various  management tools, including student  records, course records, financial  records, and event records.  2. The system should provide  administrators with the ability to  manage university resources,  including facilities, equipment, and  technology. 3. The system should support various  administrative tasks, including staff  management, resource allocation, and  policy development.  4. The system should provide real-time  reporting and analytics, including  performance metrics, financial  reports. | 1. The system should allow students  and faculty to view and schedule  transportation services, including  shuttle buses, taxis, and car rentals  2. The system should allow students  and faculty to track the real-time  location of transportation services,  including estimated arrival and  departure times. | 1. The system should allow event  organizers to create and manage  events, including setting event details,  dates, times, locations, and attendee  limits.  2. The system should allow attendees to  view event details, register for events,  and manage their event registration  status.  3.  The system should support various  event types, including seminars,  workshops, conferences, and social  events  4. The system should allow organizers  to set event fees and collect  payments from attendees, with a  range of payment methods. | 1. The system should allow students to  view their current account balance  and transaction history.  2. The system should allow students to  make payments towards their account  balance using various payment  methods such as credit/debit cards,  bank transfers, or e-wallets.  3. The system should validate the  payment information and provide  confirmation to the student upon  successful completion of the  payment. |



|---------------------------------------------------------------------------------------------------------------------------------------|

EERD MODEl :

Entities :

* Student: Stores information about each student including their name, student ID, username, password, and other personal details.
* Professor: Stores information about each professor including their name, professor ID, username, password, and other personal details.
* Course: Stores information about each course including course code, course name, course description, and the professor who teaches the course.
* Semester: Stores information about each semester including start date, end date, and semester ID.
* Enrollment: Stores information about each student's enrollment in a course including the student ID, course code, semester ID, and other enrollment details.
* Grade: Stores information about each student's grade in a course including the student ID, course code, semester ID, coursework marks, final exam marks, and final grade.
* Fees: Stores information about each student's fees for a semester including the student ID, semester ID, registered courses, fees amount, and payment status.

Relationships:

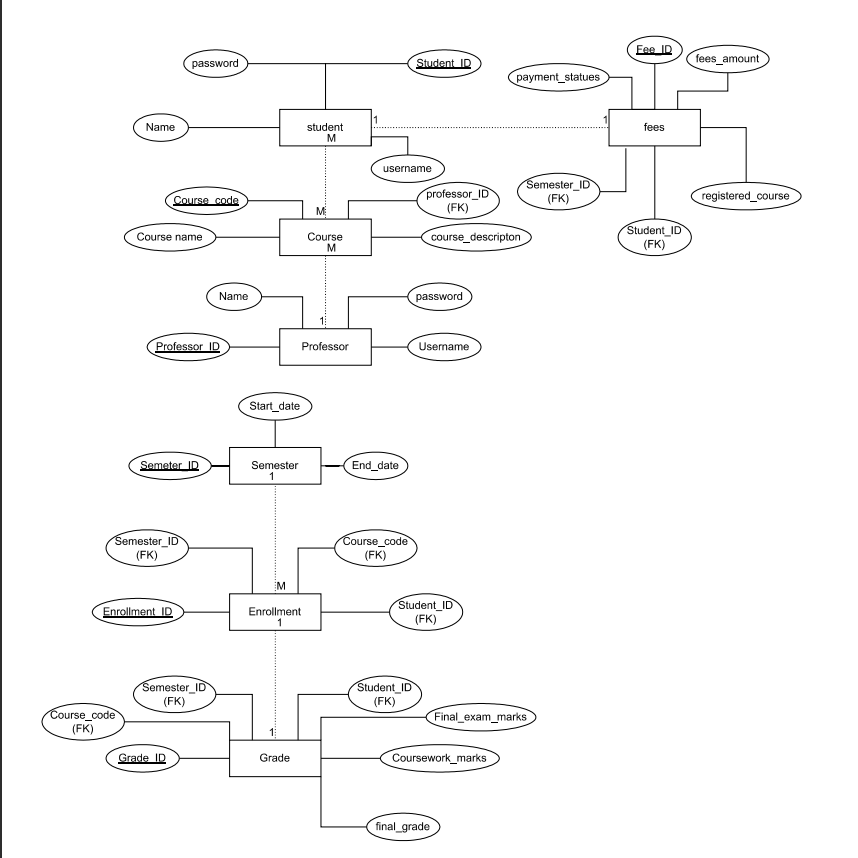
* Each student can enroll in many courses, and each course can have many students enrolled in it. This is a many-to-many relationship between Student and Course, which is resolved by the Enrollment entity.
* Each course can be taught by only one professor, but each professor can teach many courses. This is a one-to-many relationship between Professor and Course.
* Each enrollment is associated with one semester, and each semester can have many enrollments. This is a one-to-many relationship between Semester and Enrollment.
* Each grade is associated with one enrollment, and each enrollment can have one grade. This is a one-to-one relationship between Enrollment and Grade.
* Each fee is associated with one student in one semester, and each student in one semester can have one fee. This is a one-to-one relationship between Student and Fees.

Schema:

* Student: student\_id (primary key), name, username, password, and other personal details.
* Professor: professor\_id (primary key), name, username, password, and other personal details.
* Course: course\_code (primary key), course\_name, course\_description, professor\_id (foreign key).
* Semester: semester\_id (primary key), start\_date, end\_date.
* Enrollment: enrollment\_id (primary key), student\_id (foreign key), course\_code (foreign key), semester\_id (foreign key), and other enrollment details.
* Grade: grade\_id (primary key), student\_id (foreign key), course\_code (foreign key), semester\_id (foreign key), coursework\_marks, final\_exam\_marks, final\_grade.
* Fees: fee\_id (primary key), student\_id (foreign key), semester\_id (foreign key), registered\_courses, fees\_amount, payment\_status.Course.
* Each enrollment is associated with one semester, and each semester can have many enrollments. This is a one-to-many relationship between Semester and Enrollment.
* Each grade is associated with one enrollment, and each enrollment can have one grade. This is a one-to-one relationship between Enrollment and Grade.
* Each fee is associated with one student in one semester, and each student in one semester can have one fee. This is a one-to-one relationship between Student and Fees.

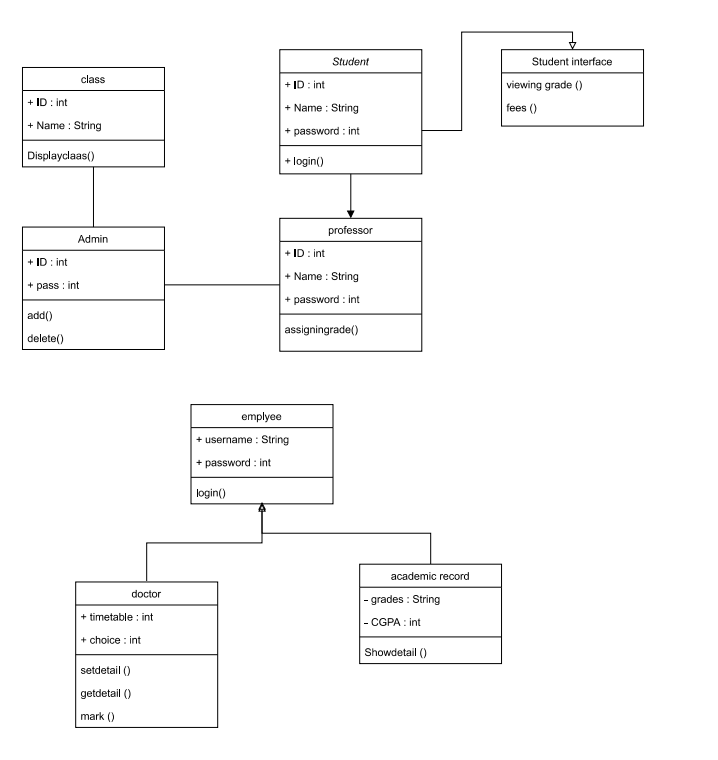
EERD MODEl :

Link: 



Class diagram :

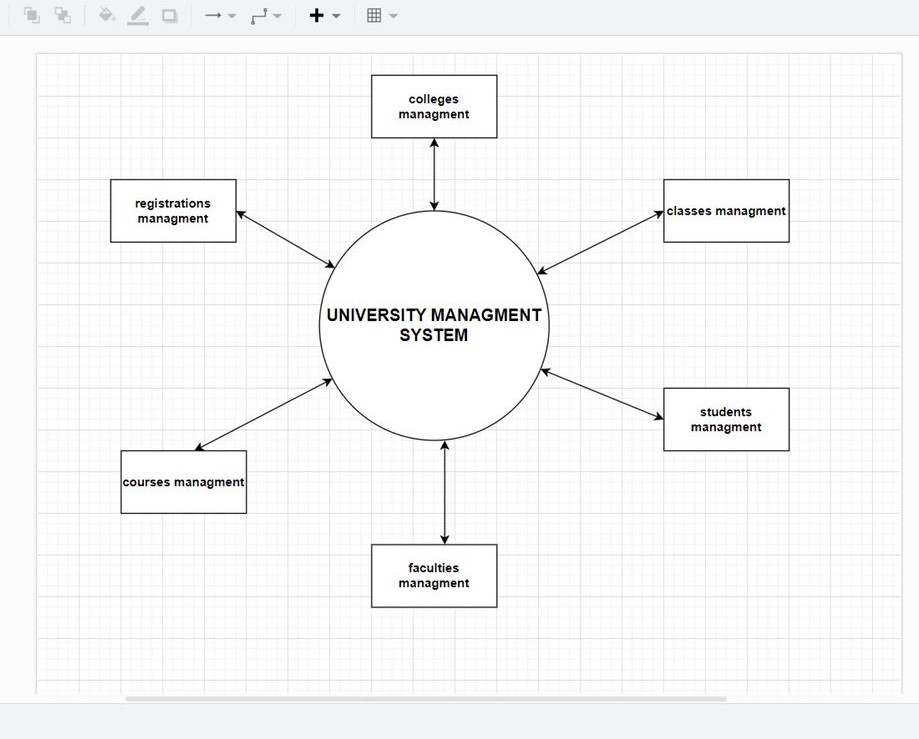




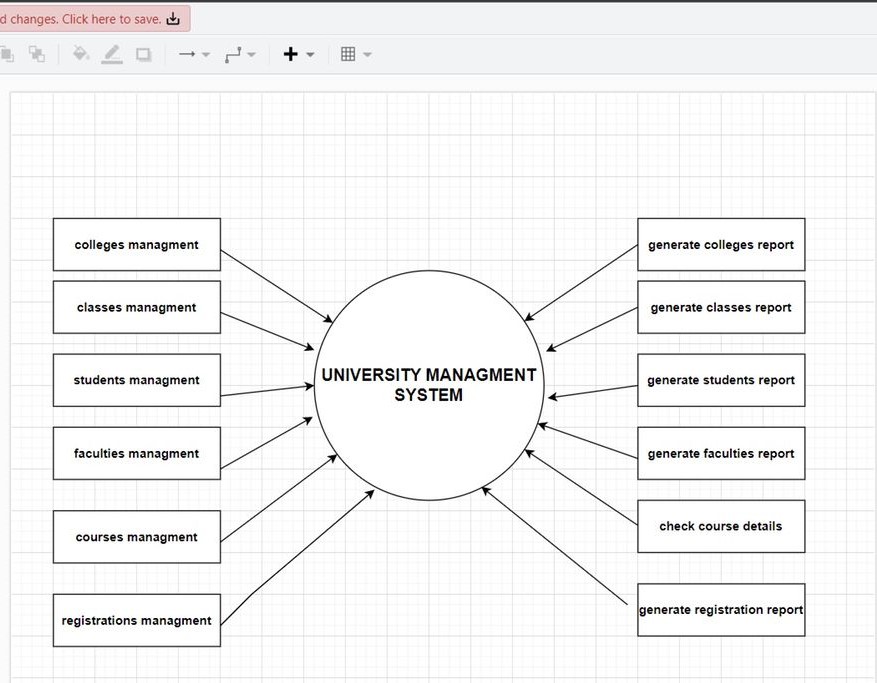
|  |  |  |
| --- | --- | --- |
| **USE case** | **Desciption** | **classes** |
| UC-01 | login | User , Student , Professor , HRPersonnel , Admin |
| UC-02 | Logout | User , Student , Professor , HRPersonnel , Admin |
| UC-03 | View course catalog | Student |
| UC-04 | Register for course | Student , course |
| UC-05 | View enrolled courses | Student |
| UC-06 | view course schedule | Student |
| UC-07 | View grades | Student |
| UC-08 | View fees | Student |
| UC-09 | Offer course | Student |
| UC-10 | View enrolled students | Professor , course |
| UC-11 | Assign grades | Professor , course |
| UC-12 | Search professors | Professor , grade |
| UC-13 | Professor salaries | HRPersonnel |
| UC-14 | Add user | Admin |
| UC-15 | Delete user | Admin |

Data Flow diagrams :

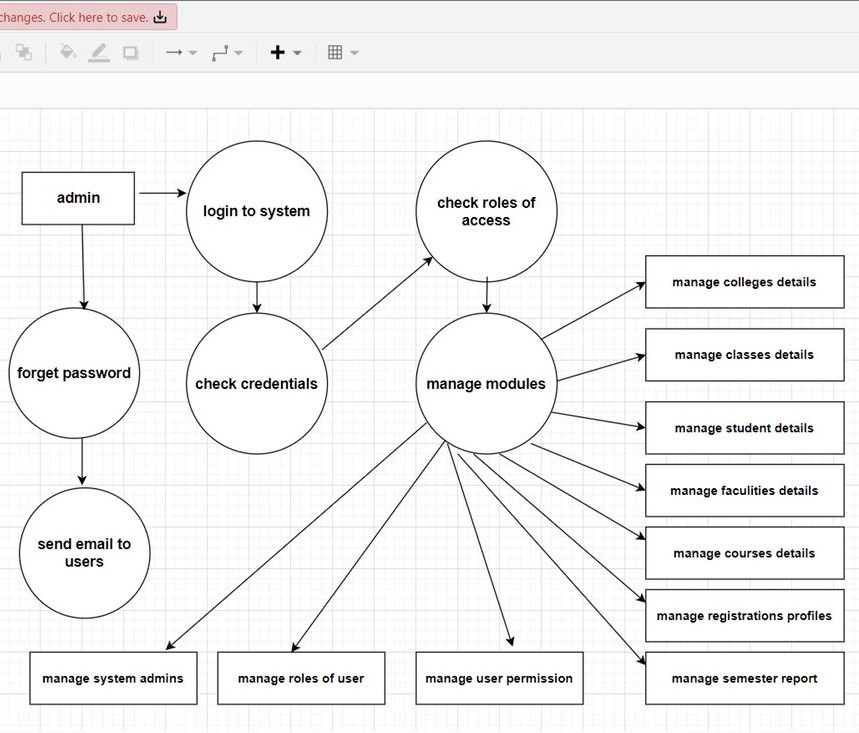
Level 0 :



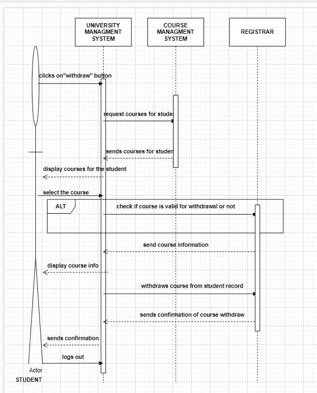
Level 1:

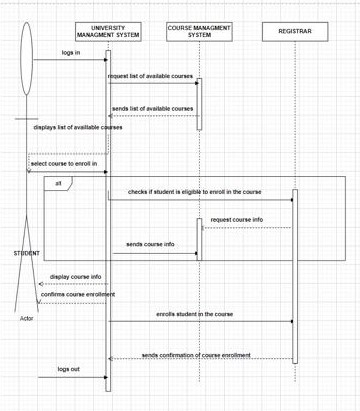


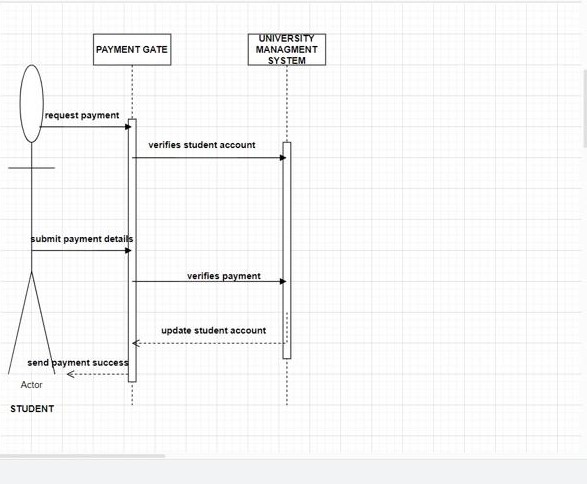
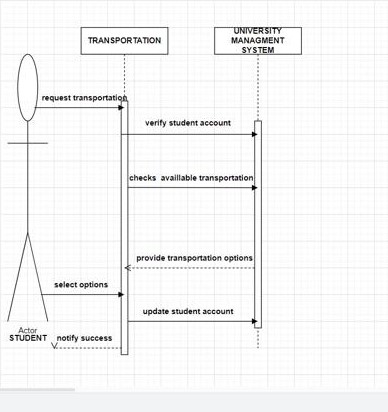
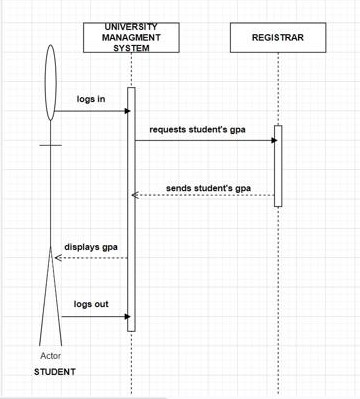
Level 2:



The sequence diagrams :







Interface :

* UserInterface: This interface defines the common methods for all types of users, such as logging in and logging out.
* StudentInterface: This interface defines the methods specific to students, such as registering for courses, viewing grades, and viewing fees.
* ProfessorInterface: This interface defines the methods specific to professors, such as offering courses, viewing enrolled students, and assigning grades.
* HRInterface: This interface defines the methods specific to H.R personnel, such as searching professors and calculating their salaries.
* AdminInterface: This interface defines the methods specific to admins, such as adding and deleting users from the system