CSE 251 Software Engineering
University Management System

System that manages and organize all university aspect to
students and staff

Manager:
Ayten Hossam id:21100785

Team members:
Aysha mohamed id:21100798
Ahella mohamed id:21100824
Hana mohamed id:21100863
Ahmed Elsayed id:21100818

Customer Requirements:

I. *AS a new student:*

1) A system that you can register on as a new student with typical registration
requirements with fast response.
2) A brief of each faculty and good representation on them.

II. *As new employee:*

1) A system that you can register as a new employee with typical registration
requirements with fast response.
2) Show available different positions that you can choose from.

III. *As an existing student:*

1) ADDING OR DROPPING COURSES: enrolling into courses and if wanted dropping
any course at anytime

2) GUIDNESS TRANSCRIPT.

3) CALCULATING EACH SEMESTER'S GPA: when each semester is done when
receiving final grades system should provide the GPA for the semester.

4) GPA CALCULATOR: if the student would like to do his own calculations on
what he expects to get.

5) Clinic reservation: there should be an option to reserve a clinical appointment to avoid being crowded.

6) Invoices on each semester payment with online payment methods: the system should provide the invoice on each payment of every semester.

7) Questioner: Adding an option for the student to ask and interact with the university staff.

8) Library access: Giving the student the access for Library from books to whatever he wants and showing their availability.

9) Keeping track of students' performance: The student should be provided his time-by-time performance to keep track of his marks and to have easy access to them.10)

Academic Entertaiment news: on every entertainment related news should also be provided.

11)

Student appeal: If the student would like to make sure that the final mark, he got is really correct.

12)

Work submissions: An option that allows the student to submit his work and reviewing it to doctors / TA.

13)

Bus booking: At the beginning of each semester there should be a bus booking option to make it easy for the student.

14)

Final exam excuses: If there were any real-life excuses for the student to not to attend the final exam so there should be an option to put on these kinds of excuses.

15)

STUDENT PROFILE AND INFO: Giving access for the student's profile.

16)

SETTINGS: allowing the student to change any info related to his account like password.

17)

TECHNICAL SUPPORT: In case of any technical issues students should be
provided the help for it.

18)

CONTACT THE OFFICE: In case of wanting to ask about any questions regarding the university.

IV. ***As an existing employee***:

1) **As a doctor**: You can view your student's data and adding their performance/editing them. View Salary history and raises. View attendance
on lectures. Dropping courses for students. Giving students grades and updating them with their performance based on the last marks.

2) **As a Teaching assistant**: You can view your student's data and adding their
performance/editing them. View Salary history and raises. View attendance
on lectures. Dropping courses for students. Add lab performance and answer students' question on questioner.

3) **As a department member**: Having access to all university members'
accounts but, answering to any problems facing any user, accepting new registered students based on the requirements, accepting who applied for
available jobs, reading and accepting students appeal and then sending them to their proper doctors.

4) **Staff**: Viewing their salary/raises history and attendance on work days.

Subsystem we have in this system:

We have 8 subsystems:

Loging in Management: It is a system for all different users to enable them to log in to the system with the provided username and passwords.

Salary Management: It is a sub system for doctors and teaching assistants A) that automatically update them with their monthly salary B) shows their salary history.

Student Record Management: is a subsystem for students to view their A) performance in case of showing attendance and grades in general in the course they chose to view their performance in B) view final grades of the past semester when the student chooses the semester, they would like to view the grades in a report appears with CGPA and term GPA and course codes with total credit hours of semester and credit hours of each course along with grades.

Courses Management: Is a subsystem for students that has A) adding and dropping courses for student, the student chooses the course he would like to add or drop then the system

automatically removes or add it. B) after that can view their
timetable C) can also view the guideness transcript for the courses.

Payment options Management: it is a subsystem for students
that A) Has an option of printing the invoice of the semesterAnd paying by the bank. B) Or paying online with Mastercard.

Class Material Management: is a subsystem made for the
students to view the course material that the doctor/TA has
added.

System management: subsystem that the department handles
and A) has access to all users B) sending invoices to students
and C) salary for employees D) Creates and send s the reports of
each semester to the students.

Courses in progress Management: is a subsystem for the
courses that the doctor has, he has access to add material,
drooping courses for some students that their performance
didn't satisfy the needed mark, add attendance/marks and
being able to edit them.

The Manual Script for each User:

1.Once the website is open the user can log in to an existing account or type in his username and password that was given by the university department.

1.1.the user have the option to use google password manager for easy access to his account

1.2. For our system according to each account the user receives his requirements as (Student,TA,Doctor,….)

1.3. Upon logging successfully, the user will get his desired requirements.

1.4. The user now will be able to make his choice.

2.This is the student manual script.

2.1. Firstly, a tab will appear with 5 different options **[Courses button, Class material button, Record button and payment button, training field.]**

2.2.1. if the student chooses **Courses button**

- a tab will appear with 3 options **(Add/Drop classes button, Guide transcript button and view timetable button)**

2.2.1.1. **In case of choosing Add/Drop classes.**

- A tab will appear with two buttons (Add class/Drop class)
- The student chooses one of them.

A) In case of choosing Add class:1)The system provides the user with all the courses that are available
and he can enroll to (according to the semester he is in and his GPA)

2)Then the student chooses which course he would like to enroll to.

3)Upon choosing the course, a tab will appear with all available timings for the course.

4) Then the student chooses his suitable timing.

5) After that a message will appear that he enrolled successfully.

B) In case of choosing Drop class:

1)The system provides the student with all of his already enrolled to courses

2)Then the student makes his discussion of dropping which class.

3)A successful message appears of dropping this course

2.2.1.2. **In case of choosing Guide Transcript**

- A tab will appear with a semester from 1 to …...
- The student chooses the semester that he is about to be in
- A list of courses appears and that would be the system giving him some sort of an example of what to take in this semester to help him with adding the courses in this semester.

2.2.1.3. **In case of choosing View Timetable**

- A tab will appear with a table that shows all courses student has enrolled to and classified throughout the week with the proper timing of each course's(Lecture,Lab,Tutorial,…...) and total number of courses enrolled to.

2.2.2. If the student chooses class **material button**

- a tab with the option of viewing every class the student enrolled in
- The student can choose any course to open.
- another tab to view the class material: lectures (slides, videos) Tutorial (slides, videos), Quizes answer.

2.2.3. In case of choosing **Record button**.• A tab will appear with two buttons **(Performance/Semester past Grades)**

- The student chooses one of them.

A) In case of choosing performance:

1)The system provides the student with a list of his already enrolled to courses.

2)Student chooses the course he would like to view his performance in.

3)A tab will appear with a table containing all his semester work added on by the doctor or TA throughout the whole semester.

B) In case of choosing Semester past Grades:

1)A tab will appear with a list of all the semester that the student passed and the system calculated his GPA in (either **CGPA or GPA**)

2)The student chooses which of the semesters he would like to check on his marks.

3)A report will appear with:

- Each course name and the hours for it to be completed and its grade.
- The total GPA of the semester only.
- The CGPA of all the semesters passed
- The total credit hours he took in the semester.

2.2.4. If the student chooses **Payment button**,

- a tab will appear with 2 options **(Online payment button and print invoice button)**

A) In case of choosing online payment button

- a tab will appear with different methods to choose from to pay student invoice (visa card, master card, apple pay, fawry pay...)
- a tab will appear with card information so the student can fill

them and the bill will appear afterwards

B) In case of choosing print invoice button

• a tab will appear with student invoice and print button options so the student can print it as receipt.2.2.5. if the student chooses **field training button**

2.2.5.1. a tab will appear with 4 different options**(add training,drop training, view schedule,view materials)**

2.2.5.2. if his choice is **add training.**

• A tab will appear with different training that the student can choose from to enrol.

2.2.5.3. if his choice is **drop training.**

• A tab will appear with the training that the student only enrolled in and he can drop what he want

2.2.5.4. if his choice is **view materials.**

• a tab will appear with training materials.

2.2.5.5. if his choice is **view schedule.**

• A tab will appear with training schedule.

3.This is the Doctor's manual script.

3.1. Firstly, a tab will appear with 2 different options (**Courses in progress/Salary**)

3.1.1. If his choice is **courses in progress**

• The system shows a list of all courses that the doctor gives this semester.

• The doctor chooses which course he would like to view.

• it will appear the tab that has 4 options (**adding/view materials, dropping courses, adding/edit marks and attendance, show students enrolled**)

3.1.1.1. If his choice is **Adding /view materials:**

• doctors and TA have access to upload their lectures,recordings,tutorials and assignments.

3.1.1.2. If his choice is **dropping courses:**

• A list of all students appears.

• The doctor chooses which student that his performance wasn't enough for the course• Then a confirmation message appear that the student now is no
longer in the course.

3.1.1.3. If his choice is **adding/edit marks and attendance:**

• the doctors have sheet about student's name,ID and each student absence and presence in the past lectures.

• The doctor has ability to put

(project,midterms,exams,quizes,assignments,semester work,attendance) marks and edit them whenever he needs to.

3.1.1.4. If his choice is **showing students enrollment:**

- A list of students who have enrolled in the course will appear.

3.1.2. if the doctor chooses "**salary**"

- New tab with 2 options will be appear (**View my salary, salary history**)

A) if he chooses View my salary:

- a blank tab will be appeared with his monthly salary.

B) if he chooses salary history:

- a tab will be appeared with all months from he started to work till now.
- then he will choose the month that he wants to know the salary of.
- after choosing the months the blank tab with his salary will be appear

4.This is the TA's manual script.

4.1. Firstly, a tab will appear with 2 different options (**Courses in progress/Salary**)

4.1.1. If TA choice is **courses in progress**

- The system shows a list of all courses that the TA gives this semester.
- The TA chooses which course he would like to view.
- it will appear the tab that has 4 options (**adding/view materials, adding/edit marks and attendance, show students enrolled**)3.1.1.1. If TA choice is **Adding /view materials:**
- TA have access to upload their lectures,recordings,tutorials and assignments.

3.1.1.2. If his choice is **adding/edit marks and attendance:**

- the TA have sheet about student's name,ID and each student absence and presence in the past lectures.
- The TA has ability to put (project,midterms,exams,quizes,assignments,semester work,attendance) marks and edit them whenever he needs to.

3.1.1.3. If TA choice is **showing students enrollment:**

- A list of students who have enrolled in the course will appear.

4.1.2. if the TA chooses "**salary**"

- New tab with 2 options will be appear (**View my salary, salary history**)

A) if he chooses View my salary:

- a blank tab will be appear with his monthly salary.

B) if he chooses salary history:

- a tab will be appeared with all months from he started to work till now.
- then he will choose the month that he wants to know the salary of.
- after choosing the months the blank tab with his salary will be appear

5.This is the Department manual script.

5.1. Firstly, a tab will appear with 3 different options **[Access to all accounts, sending salary, sending invoices, Sending /creating Reports of each semester}**

5.2.1. if the Department chooses **Access to all accounts**

- **A list with choosing either(student,doctor,TA)**
- **Then a tab with a list of all account appears**
- **The department can edit or do whatever he wants**5.2.2. if the Department chooses **Sending salary**
- **A tab will appear with two icons doctor or TA**
- **Then a list of all Doctors or TA**
- **Then the department chooses who he would like to send**
- **And press the upload button to upload the bill**

5.2.3. if the Department chooses **Sending invoices**

- **A tab will appear with all student names**
- **The department chooses which to send to**
- **Then press upload button to upload the invoice**

5.2.4. if the Department chooses **Sending/Creating Reports**

- A tab will appear with two options Create or send

5.2.4.1. **If the department chooses create:**

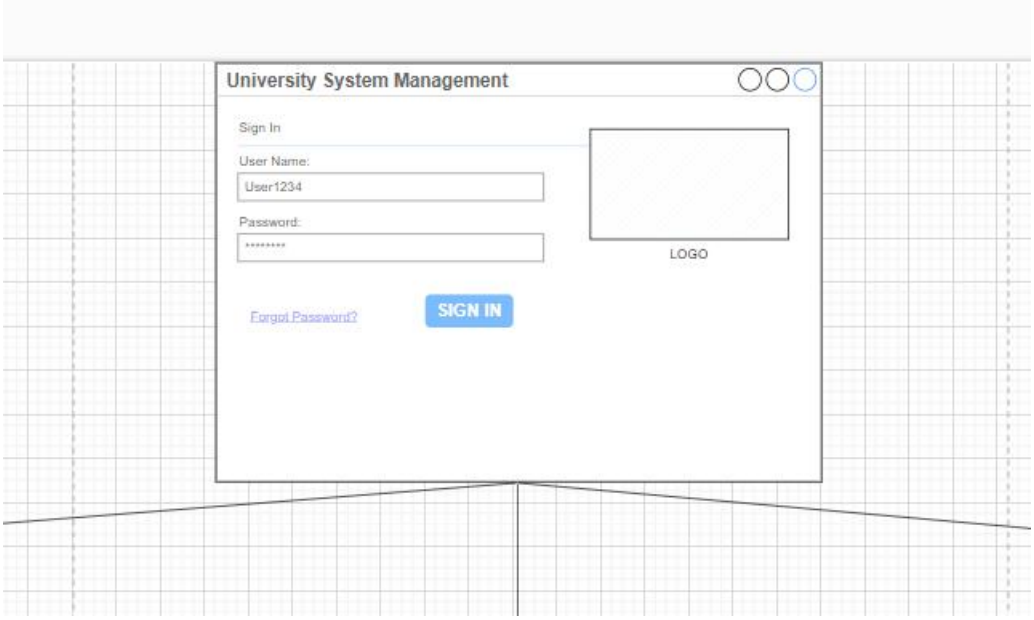
- **A blank report page appears and he has to fill in the blanks with student information.**
- **Then he fills in the courses with grades and credit hours and points**
- **The system already calculates the CGPA and the GPA according to the courses added by Department.**

5.2.4.2. **If the department chooses send:**

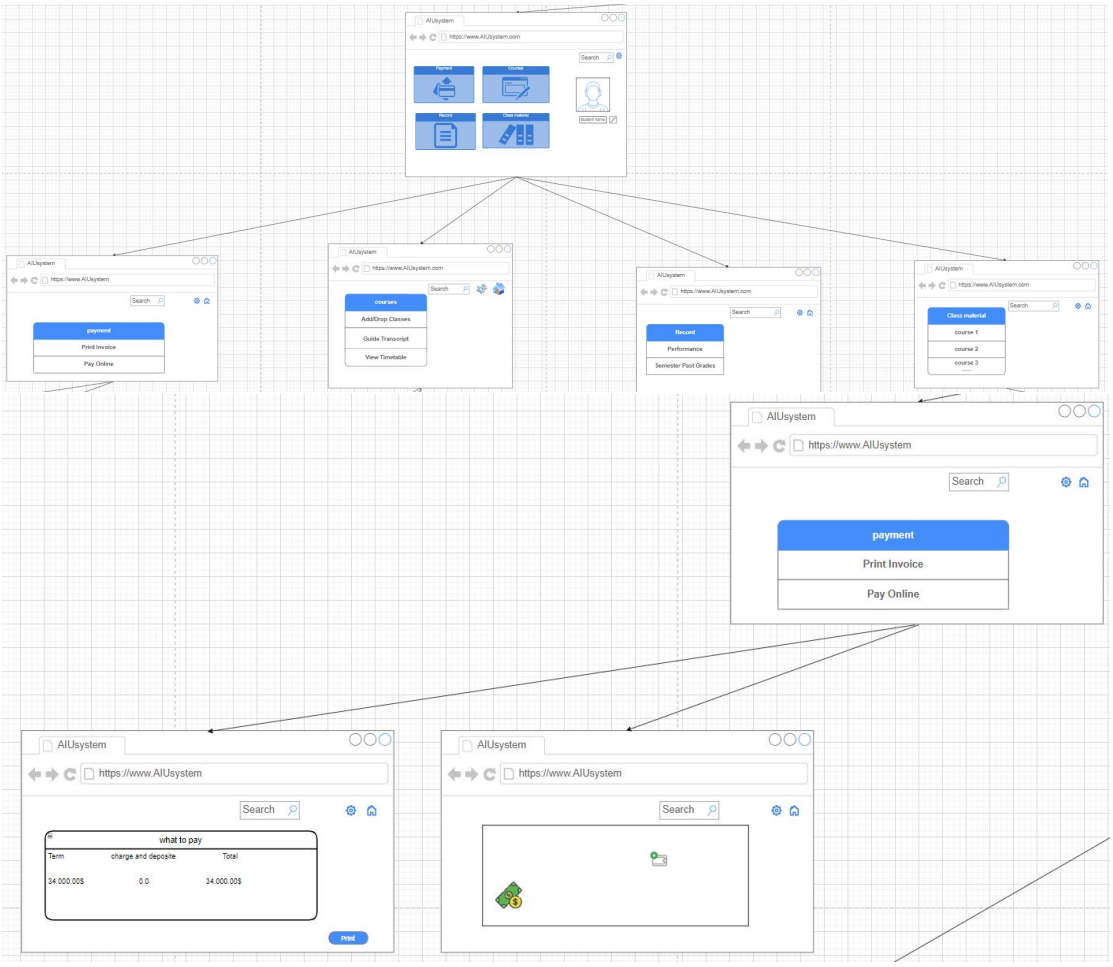
- **A list with all students appears.**
- **The Department chooses.**
- **Then sends.**

System GUI:

Login Tab:

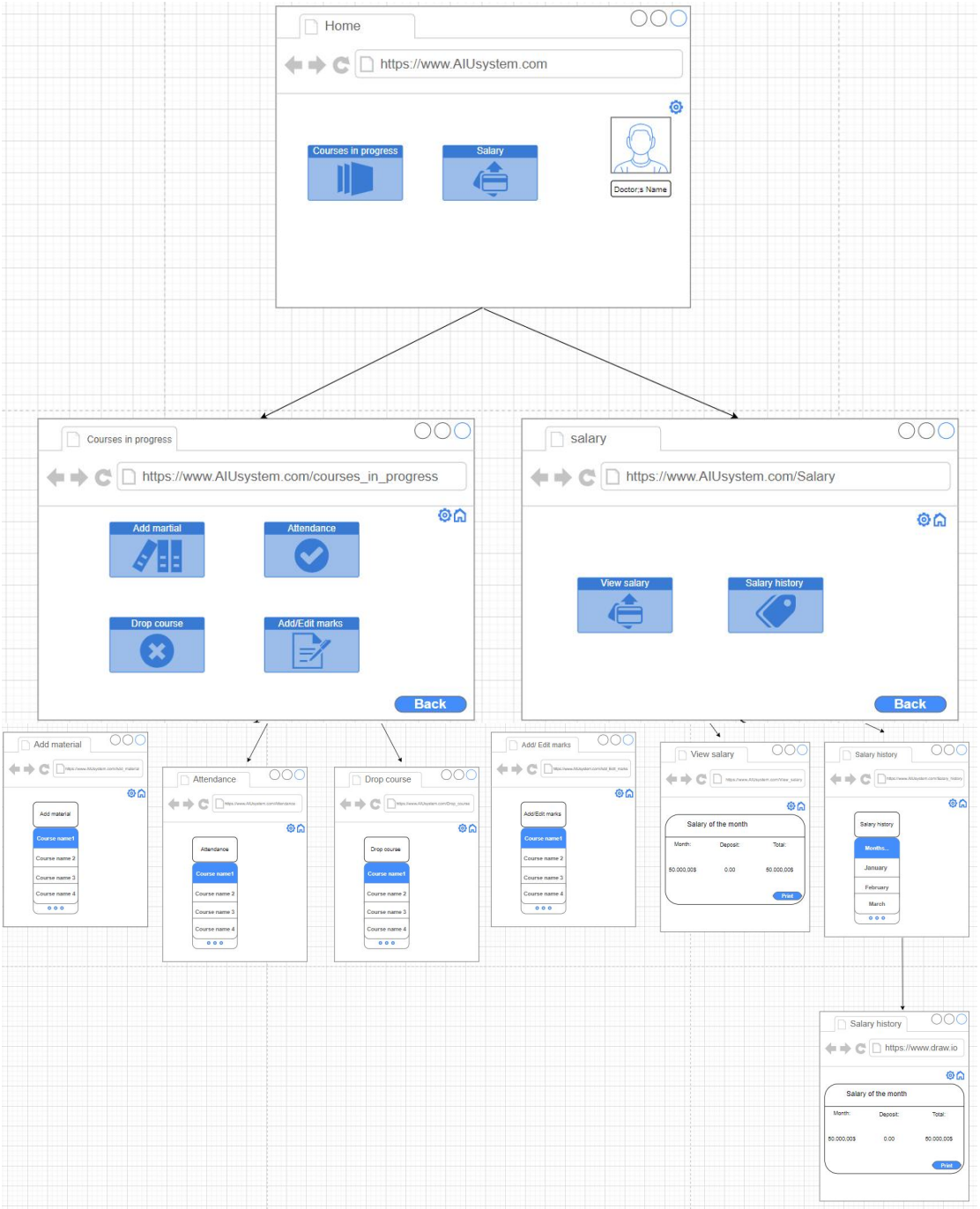


Student Tab:

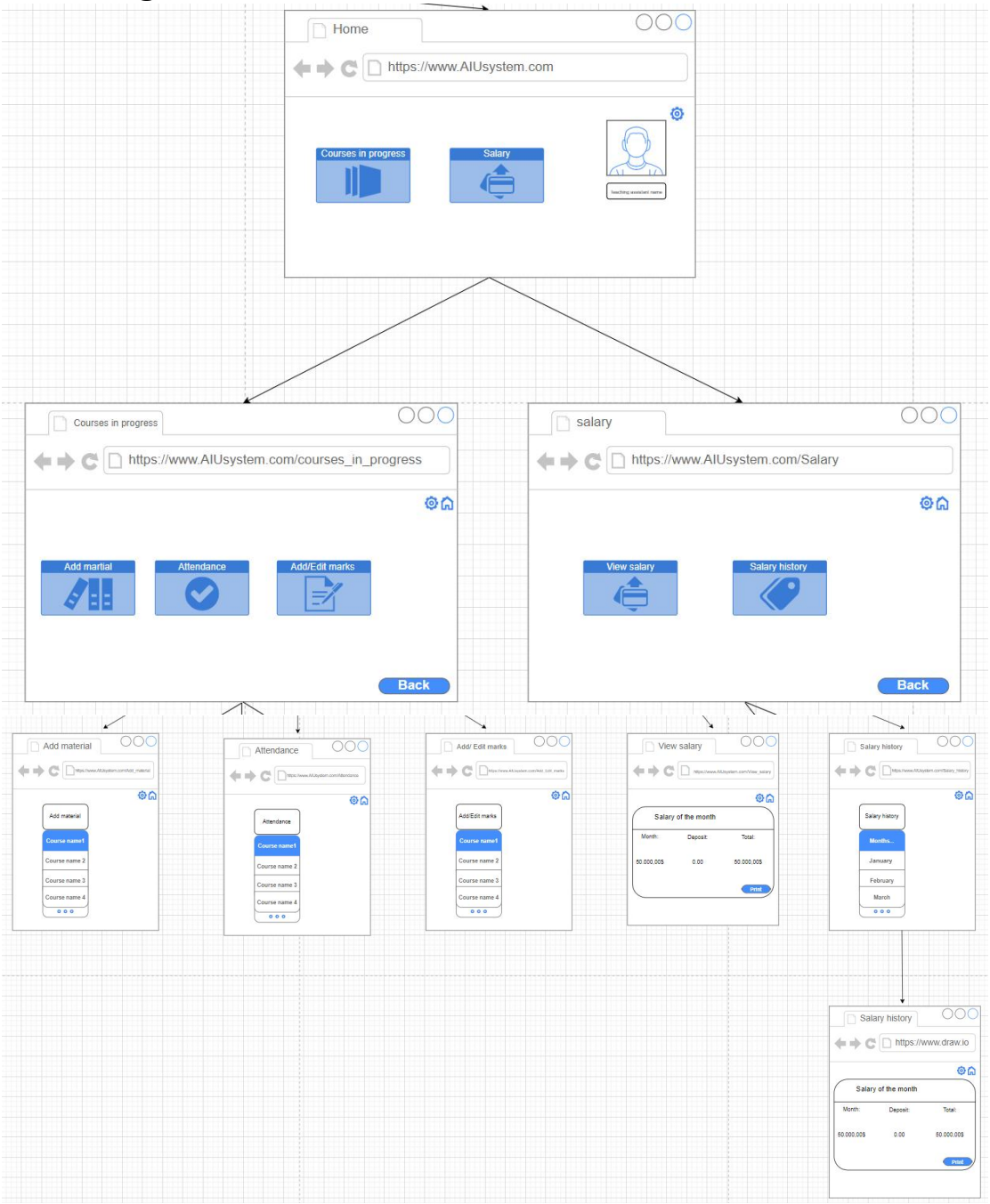

AIUsystem
https://www.AIUsystem
Search
payment
Print Invoice
Pay Online
what to pay
Term
charge and deposite
Total
34.000.00$
0.0
34.000.00$
Print

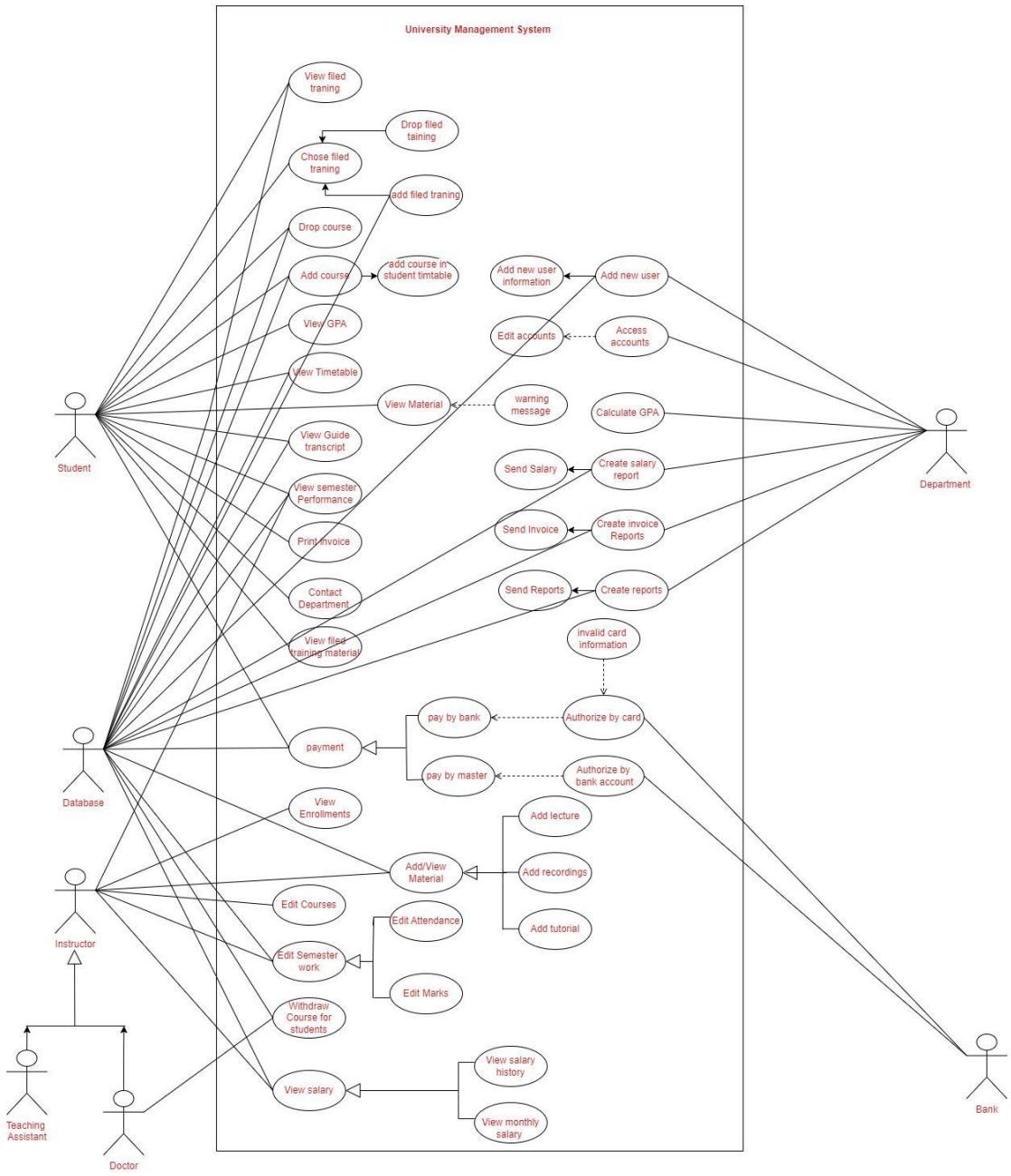

AIUsystem
https://www.AIUsystem.com
Search
Add course
Course one
Course two
Course three
Drop course
Course one
Course two
Course three
choose one of the courses
course1
course2
couse3
choose the semester
spring--2022
fall--2022
spring--2023
View Timetable
Table
Sun CSE121 AIE111
Mon MAT212 MAT111
Tues CSE121 LAN212
Record
Performance
Semester Past Grades
Class material
course 1
course 2
course 3
course 1
Lecture slides/videos
Tutorial slides/videos
Quizzes answers

Doctors Tab:


Home
https://www.AIUsystem.com
Courses in progress
Salary
Doctor;s Name
Courses in progress
https://www.AIUsystem.com/courses_in_progress
Add martial
Attendance
Drop course
Add/Edit marks
Back
salary
https://www.AIUsystem.com/Salary
View salary
Salary history
Back
Add material
Attendance
Drop course
Add/ Edit marks
View salary
Salary of the month
Salary history
Salary history
https://www.draw.io
Salary of the month

Teaching Assistant tab:




University Management System
View filed traning
Drop filed taining
Chose filed traning
add filed traning
Drop course
Add course
add course in student timtable
View GPA
View Timetable
View Material
View Guide transcript
View semester Performance
Print invoice
Contact Department
View filed training material
payment
View Enrollments
Edit Courses
Edit Semester work
Withdraw Course for students
View salary
Add new user information
Add new user
Edit accounts
Access accounts
warning message
Calculate GPA
Send Salary
Create salary report
Send Invoice
Create invoice Reports
Send Reports
Create reports
invalid card information
pay by bank
Authorize by card
pay by master
Authorize by bank account
Add/View Material
Add lecture
Add recordings
Add tutorial
Edit Attendance
Edit Marks
View salary history
View monthly salary
Student
Database
Instructor
Teaching Assistant
Doctor
Department
Bank

- **The traceability matrix:**

Features	Salary Management	Student Record Management	Courses Management	Payment options Management	System management	Courses in progress Management	Class material Management	Field training
Adding or dropping courses			✓					
Guidness transcript			✓					
Calculating each semester GPA		✓						
Tracking of students' performance		✓						
Invoices on each semester				✓				
Adding materials						✓		
View class material							✓	
Department sending invoices					✓			
Viewing final student grades		✓						

Doctor/TA updating students' performance						✓		
View Salary Salary/Payment history	✓							
Access to all members by department					✓			
Sending invoices to students and salary bills to TA/DR					✓			
Adding field training								✓
Dropping field training								✓
View field training materials								✓

View field training schedule								✓

The system rules:

1. The system should have an internet connection for the user to use the information system on any browser.
2. The system should have a database in order to store all information that has been uploaded or added by the users or department and retrieving it when ever wanted alongside with the user account detail such as username and password and account information.3. The system should present all the private information such users accounts information or accessing it only by the admin which in our case is the Department.
4. The system should have a connection with the bank and authorizing abilities for all money transactions.
5. The user using the system which in our case is the student/TA/doctor should have
5.1. A bank account in order to transfer the monthly salary to it which is for the Doctor and Ta.
5.2. A bank account or a master card for money transactions of the student semester payment which the student should have
6. The user should turn on notifications for any updates that will be added.

Functional requirements:

1.The website shall have a home page that lists the purpose of the organization.
2.The website shall have a page called "**user information**" which displays the information of the user such as username, email used, user ID and all information stored in the database.
3.The website shall have an option for the department user to add in invoices for each student the system is supposed to store these invoices in the system database and then the department user selects the invoice for each student and send it to the student.
4.The website shall have an option for the department user to add in salary reports for each employee and the system is supposed to store in these reports in the system database and the department user selects the salary report for each employee and send it to the employee.
5.The website shall have an option for the department user to create the student final semester record which includes adding in final grades of the past semester and calculating in CGPA and term GPA and adding in course codes with total credit hours of semester and credit hours ofeach course along with grades system is supposed to store in these reports in the system database and the department user selects the

student report for each student and send it to the student.
6.The website shall have an option for the doctor/TA user to add in the materials of the courses they are teaching after choosing the course desired to add in material from a list of courses TA/Doctor are teaching this semester and the system function is to store these materials in the system database and display it for student whenever he wants to view them by choosing the course from his account from viewing class material.
7.The website shall have an option for the Doctor to drop a certain student who didn't satisfy the course marks by selecting the course name first that he teaches then selects the student and the system responsible for removing this student from this relation in database and removing the course from the student account.
8.The website shall have an option for the Doctor/TA to add in information of the student in a certain course that they choose from a list of the courses they teach and the system responsible to add these data to the system database and display it for student whenever he wants to view them by choosing the course from his account from viewing record.
9.The website shall have an option for the student user to add / drop courses at a certain time maximum in week 7 these courses can be dropped or added
9.1. In case of adding courses, the student will choose the wanted course and the system will add student in this course relation in database.
9.2. In case of dropping courses, the student chooses the wanted course to drop and the system is responsible for removing this student from this relation in database and removing the course from the student account.
10.The website shall have the option for the student to view the timetable of the semester that has been already made by the system and added to the student account.11.The website shall have a home page that lists the purpose of the
organization.
12.The website shall have a page called "payment" which displays all information payment of the student that stored in the system database.
13.The website shall have an option for the student to print the invoice for the current semester that has been added in and saved in the database by the department.
14.The website shall have an option for the student to pay through the

bank, the system is responsible for authorizing the bank account information and taking the amount of money needed.
15.The website shall have an option for the student to pay with Master card, the system is responsible for authorizing the bank account information and taking the amount of money needed.
16-The website shall have an option for the doctors/TA to view the performance of the students showing their attendance, participation and grades in general in courses they chose for each student and the doctors/TA selects the performance report for each student and view the performance to the students.
17-The website shall have an option for the student to view final grades of the past semester when the student chooses the semester, they would like to view the grades in a report appears with CGPA and term GPA and course codes with total credit hours of semester and credit hours of each course along with grades.
18-The website shall have an option for the doctors/TA to add materials and add the material to the system database to make the students able to view material that has been added by the doctors/TA.
19-The website shall have an option for the student to view all class material which is the course material that has been added in by the doctor/TA by selecting the icon of class material and choosing the desired course.
20-The website shall have an option for the Doctor/TA to view the monthly report salary which has been added and sent by the department and stored in the database.21-The website shall have an option for the student to view their
performance that has been added by the doctor/TA in every course he has enrolled on, and all this information is stored in database.
22-The website shall have an option for the department to add in new student to the information system or a new Doctor/TA by storing in their information as in the national id and full name and their CV and then the system provides for the department the given user name and password for the new user.
23-The website shall have an option for the department to have access to all users using this information system and it happens by entering the user name of the user he would like to access from a big list that the department has and then the department can edit and check on the account.
24-The website shall have an option for the student to enrol field training and choose from a list includes different training and the system

function is to store each student data that enrolled in system database or to drop the field training and the system function remove the student data from field training database
25-The website shall have an option for the student to access field training materials and the system function is to store the field training materials that has already been added.
26-The website shall have an option for the student to view field training schedule and the system function is to store the field training schedule.
27-The website shall have an option called “Guideness transcript “which displays the suggested courses to the students according to their GPAGPA which has been added already to the system database.
28- The website shall have an option called “salary history “which displays the salary history information for doctors and teaching assistants which are all stored in the system and sent to them in the past months by the department who already made the reports and stored them in the system databaseThe website shall have an option for ddoctors and teaching assistants to
view their salary history which is all stored in the system database.

Nonfunctional requirements:

1.Performance requirements
1.1. The system must respond the business operations in less than 3 seconds for the user (student, doctor, TA) and department.
1.2. The system should be compatible with all modern browsers such as google, safari and more so its access by the user from more than one different browser.
1.3. The system should respond to the operation messages to the user within 2 seconds.
1.4. The system should be reliable and well organized for the user to know where exactly everything lies.
1.5. The system should authorize and verify the user’s bank account or master card in less than 5 minutes.
1.6. The system must take a reasonable amount of time when it comes to uploading materials that depends on the size of it.
1.7. The system must handle the transactions of materials such as (salary reports, semester reports) in less than 2 minutes.
2.Security requirements/ Safety requirements
2.1. The database should be secured from SQL injection and other external sources to prevent leak of information
2.2. The system could use security packages or layers to secure the

data being transmitted and added in by the users.
3.Reliability
3.1. The program must be reliable.
3.2. The program must provide crashes exceptions so that unintended results do not occur such as system crashes or data validation failures.
4.Usability4.1. The system should have user-friendly interfaces for the user to
easily access it.
5.Availability
5.1. The system must be available 24/7, with no more than 10 minutes down time per day.

Subsystem Name	Subsystem Function	Subsystem Interface
Logging in Management	It is a system for all different users to enable them to log in to the system with the provided user name and passwords.	Public void enterusername(String username) Public void enterpassword(String password) Public boolean validateusername(String username) Public boolean validatepassword(String password)
Salary Management	It is a sub system for doctors and teaching assistants A) that automatically update them with their monthly salary B) shows their salary history.	Public void viewmonthlysalaryreport(String salaryreport) Public void viewsalaryhistoryreports(Object<historyreport> arr)
Student Record Management	is a subsystem for students to view their A) performance in case of showing attendance and grades in general in the course they chose to view their performance in B) view final grades of the past semester when the student chooses the semester, they would like to view the grades in a report appears with CGPA and term GPA and	Public void displaycoursemenu() Public void choosecourse(String coursecode) Public void viewpreformace(Object<course preformance> arr) Public void viewpastsemsterpastreport(Object<pastsemster> arr)

	course codes with total credit hours of semester and credit hours of each course along with grades.	
Courses Management	Is a subsystem for students that has A) adding and dropping courses for student, the student chooses the course he would like to add or drop then the system automatically removes or add it. B) after that can view their timetable C) can also view the guideness transcript for the courses.	Public void displaycoursemenu() Public void choosecourse(String coursecode) Public void addcourse(String coursecode) Public void dropcourse(String coursename) Public void view TimeTable(Object<coursesenrolled> arr) Public void viewgudinesstranscript(Object<suggestedcourses> arr)
Payment options Management	it is a subsystem for students that A) Has an option of printing the invoice of the semester And paying by the bank. B) Or paying online with Mastercard	Public void printInvoice(String invoice) Public void paybybankacc(int bankaccno,String password) Public void paybymastercard(int mastercardno,int mastercardexpdate,int mastercardsecurityno)
Class Material Management	is a subsystem made for the students to view the course material that the doctor/TA has added.	Public void displaycoursemenu() Public void choosecourse(String coursecode) Public void displaycoursematerial()
System management	subsystem that the department handles and A) has access to all users B) sending invoices to students and C) salary for employees D) Creates and send the reports of each semester to the students E) create the salary reports D)creates the invoices for student	Public void accessalluser(String username,String password) Public void createfinalsemsterreport(string report) Public void createsalaryreport(string report) Public void createinvoice(string report) Public void sendinvoice(String report) Public void sendsalaryreport(String report) Public void sendfinalsemesterreport(String)
Courses in progress Management	is a subsystem for the courses that the doctor has, he has access to add material, drooping courses for some students that their	Public void displaycoursemenu() Public void choosecourse(String coursecode) Public void addclassmaterial(Object<courses> arr) Public void

	performance didn't satisfy the needed mark, add attendance/marks and being able to edit them.	dropstudent(Object<students> arr, Object<courses> arr)
Field Training	Is subsystem for the student that can enroll in and choose which field training to take in each semester or drop and access field training material and view schedule	Public void displayfeildtrainingmenu() Public void choosefeiledtraining(String feiledcode) Public void enrollfeild(Object<feildoptions> arr) Public void drpfeild(Object<feildoptions> arr) Public void viewfieldmaterial(Public void enrollfeild(Object<feildoptionsmaterial > arr)

Use case 1: Create salary reports
Actor: Department
Description: The department user needs to add in the monthly salary report for each employee which he creates by adding the name of the employee and his ID and by creating this report he needs to calculate in the number of hours he has worked this month and determine part-time employee or a full-time employee and then calculate the amount of the salary after that calculate the VAT and then send it to the required employee.

Actor intentions	System responsibility
1-1-The department user opens the website through the browser he chooses.	2-1-System gets the entered user information 2-2-System validates that such user exists.
1-2-The department user enter log in page and enter username and password	2-3-System opens the dashboard page. 2-4-The system displays all options that are for the department only.
1-3-The department user selects the option of creating the salary invoice.	4-The system displays empty report for salary information for user to enter.
5-The department user fills in the empty gaps in the report by entering the employee's name, employee ID.	
6-the department user calculates the number of hours that the employee has worked this month by clicking on a tab that calculates it for him and he enters the day the employee attended and the number of hours of that day he worked.	7.The system displays the calculating work hours page. 8.The system starts to calculate the data that the department entered.

9-The department user enters the amount of the salary of the employee.	10-The system then calculates the VAT from the salary information that the department entered. 11-The system validates the information to ensure that it is complete and accurate 12-If the information is valid, the system generates a salary report with the selected parameters and displays it to the department user.
13- The department reviews the salary report and ensures that there are no errors or wrong information. 14- If everything is correct, the department user can download, print or save the report	15- The department user is returned to the system's department user menu.

- **Alternative courses:**

Step1: If there is no internet connection the system is responsible for warning the user by displaying a warning message that indicates that he needs to connect to the internet through the browser he works on.
Step 5: If the department user enters invalid information, such as an incorrect employee name or employee ID, the system displays an error message and prompts the department user to enter valid information before continuing.
Step 6: If the department user enters invalid information, such as wrong form of the number of hours then system displays an error message and prompts the department user to enter valid information before continuing.
Step 6: If the department user enters invalid information, such as an incorrect day that the employee worked on, then the system displays an error message and prompts the department user to enter valid information before continuing.
Step 12: If the system is unable to generate a salary report, such as due to a technical error, the system displays an error message and prompts the department user to try again later.

- **Test Requirements:**

1.Validate connection between the website system and the database management system.
1.1. check if there is a wrong technical connection with database
1.1.1. If so, display a technical connection error message.
1.2. check if there is a delay in the connection due to weak connection
1.2.1. If so, then the system display tries to connect with the database system message.
1.3. Validate to trying to connect again if there is wrong with connection with database2.Validate the data entered by the department in the empty gaps of the
report.
2.1. Validating the employee's name entered by the department user
2.1.1. validate that the name entered by the employee should not contain any special characters “&*#$@’\?|<>,.”
2.2.2. validate that the name contains capital letters and small letters.
2.2.3. validate that the name doesn't contain any numbers”123456789”
2.2.4. validate that there is a space between the first name and the middle name and a space between the middle name and the last name.
2.2.5. validate that there is no decimal in the employee's name that the department user entered.

2.2. Validating the employee's, ID entered by the department user
2.2.1. The employee ID should not contain any letters
2.2.2. The employee ID should only contain numbers "123456789"
2.2.3. The employee ID should be a numbered sentence with is 8 characters
2.2.4. The employee ID should not contain any special characters in it "&*#$@'\?|<>,.".
2.2.5. The employee ID should not be a negative signed number.
2.2.6. The employee ID should be already stored in the database and after the department user enters it the database displays a succeeded existing employee ID message.
2.2.7. The employee ID doesn't contain decimal number.
3.Validate the data entered by the department in the empty gaps of the calculating number of hours that the employee worked.
3.1. Validating the day name entered by the department user
3.1.1. validate that the day name entered by the employee should not contain any special characters "&*#$@'\?|<>,."3.2.2. validate that the day name contains capital letters and small
letters.
3.2.3. validate that the day name doesn't contain any numbers"123456789"
3.2.4. validate that the day name entered is an existing day which is one of seven "Sunday or Monday or Tuesday or Wednesday or Thursday or Friday or Saturday"
3.2.5. validate that the day name doesn't contain decimal.
3.2. Validating the number of hours that the employee worked entered in by the department user
3.2.1. validate that the number of hours entered by department user doesn't exceed 2 characters.
3.2.2. validate that the number of hours entered by department user is not a negative signed number.
3.2.3. validate that the number of hours entered by department user is a numbered sentence that doesn't contain any letters.
3.2.4. Validate that the number of hours entered by the department user should not contain any special characters in it "&*#$@'\?|<>,."
3.2.5. validate that the number of hours entered by the department user should only contain numbers "123456789".
3.2.6. validate that the number of hours entered by the department doesn't contain decimal.
4. Validate the calculated number of hours that the system calculated

4.1. validate that the total number of hours that the system calculated doesn't exceed 3 characters.
4.2. validate that the total number of hours that the system calculated is not a negative signed number.
4.3. validate that the total number of hours that the system calculated is a numbered sentence that doesn't contain any letters.
4.4. validate that the total number of hours that the system calculated should not contain any special characters in it “&*#$@’\?|<>,.”4.5. validate that the total number of hours that the system calculated should only contain numbers “123456789”.
4.6. validate that the total number of hours that the system calculated shouldn’t contain decimal.
5. Validate the entered salary value that the department user entered
5.1. validate that the entered salary value that the department user entered doesn't exceed 6 characters.
5.2. validate that the entered salary value that the department user entered is not a negative signed number.
5.3. validate that the entered salary value that the department user entered is a numbered sentence that doesn't contain any letters.
5.4. validate that the entered salary value that the department user entered should not contain any special characters in it “&*#$@’\?|<>,.”
5.5. validate that the entered salary value that the department user entered should only contain numbers “123456789”.
5.6. validate that the entered salary value that the department user entered is not >100000 EGP.
5.6. validate that the entered salary value that the department user entered is not <1000 EGP.
6. Validate the calculated VAT that has been calculated by the system
6.1. Validate the calculated VAT that has been calculated by the system is the real value of the 14% of the salary that the department user has entered.
6.1. Validate the calculated VAT that has been calculated by the system entered doesn't exceed 6 characters.
6.2. Validate the calculated VAT that has been calculated by the system entered is not a negative signed number.
6.3. Validate the calculated VAT that has been calculated by the system entered is a numbered sentence that doesn't contain any letters.
6.4. Validate the calculated VAT that has been calculated by the system

entered should not contain any special characters in it
“&*#$@’\?|<>,”.6.5. Validate the calculated VAT that has been calculated by the system
entered should only contain numbers “123456789”.
6.6. Validate the calculated VAT that has been calculated by the system entered is not >1000 EGP.
6.6. Validate the calculated VAT that has been calculated by the system entered is not <100 EGP.
7.Validate that all the required fields are filled with appropriate information.
7.1.Validate that username is entered.
7.1.1..Check if the username is at least 8 characters long.
7.1.2.Check if the username does not contain any special character.
7.2If not, send a warning to user accordingly.
7.2.1.Validate that password is entered
7.2.2.Check if the password is at least 8 characters long.
7.2.3.Check if the password does not contain white space.
7.2.4..If not, send a warning to user accordingly.
8.Validate the completion of the report that has been done by system that the department user receives.
8.1. Validate that all empty gaps have been filled correctly by the department user
8.1.1. If not display error message that the department has not filled in the required information.
9. Validate the saving of the report to the database.

Use case 2 : payment

Actors: Student

Description: The student needs to pay invoice of the semester.
It allows the student to select one of the two options (pay by bank or pay by Master card) in case his/her choice is pay by bank the system should make sure that the information of the user and card including their credit cardnumber, expiration date, and security code, into the payment processor's system must be correct Next, the payment processor would verify the customer's payment information and ensure that their credit card has sufficient funds to cover the purchase. it must be authorized online payment by the bank otherwise invalid card information.
In case his/her choice is pay by Master card This process involves the card holder and numerous other entities working together to complete an electronic transaction The first step in an electronic payment begins with the cardholder seeking payment with a payment card. The card holder authorizes the payment by providing it to a merchant and presenting identification if requested. After a consumer swipes his or her card through a card reader or enters the card's details in the system, the payment system sends the card's details to the merchant's bank
If a transaction cannot be authorized, it will be declined. A card could be declined for many reasons such as:
1. The card holder does not have sufficient funds in their account to cover the transaction, or the requested transaction would cause the card holder to exceed the card's credit limit.
2. The card has been reported lost or stolen.
3. The card is counterfeit.
4. The card has expired.
5. There has been a technical glitch.
6. The card holder made a mistake when entering credit card details.

Actor intentions	System responsibility
1. 1-The student opens the website through the browser he chooses. 1-2-the student enters the log in page and enter username and password	**2.2-System gets the entered user information** **2-3System validates that such user exists.** **2-4System opens the dashboard page.**
3-The student selects the option of pay invoice.	2-5--The system displays all options that are for the students only.
5. The student will tab on	4-The system displays all the

the invoice that he wants to pay. 7-The student will choose his/her way for payment 8.1.If choice is pay by bank then the student enter the bank account number and bank account password 8.2. the student enters the master card information credit card number, expiration date and Security code.	invoices that the student should be paid. 6. The system displays two option for paying (pay by bank or pay by Master card) 9.1.the system checks the bank account number and bank account password. 9.1.1.the system check that their bank account has sufficient funds to cover the purchase. 9.2.1. the system checks the master card number expiration date Security code. 9.2.2.The system check that their master card has sufficient funds to cover the purchase 10- if all information are good the system starts with drawing the amount of money needed.

	11- The system displays successful message

- **Alternative courses:**

Step1: If there is no internet connection the system is responsible for warning the user by displaying a warning message that indicates that he needs to connect to the internet through the browser he works on.

Step9: If the department user enters invalid information, such as an incorrect credit card number, credit expiration date, security code, the system displays an error message and prompts the student to enter valid information.

step10: If credit card doesn't have sufficient funds to cover the purchase, The system displays an error message and prompts the student to use another payment option to continue the process.

Step11: If The card has been reported lost or stolen or The card is counterfeit or has expired or has been technical glitch or The card holder made a mistake when entering credit card details or The card holder does not have sufficient funds in their account to cover the transaction, or the requested transaction would cause the card holder to exceed the card's credit limit ,The system displays an error message and prompts the student to use another payment option to continue the process.**Test Requirements:**

1.Validate connecting between the website system and the database management system

1.1. check if there is a wrong technical connection with database

1.1.1. If so, display a technical connection error message.

1.2. check if there is a delay in the connection due to weak connection

1.2.1. If so, then the system display tries to connect with the database system message.

1.3. Validate to trying to connect again if there is wrong with connection with database

2.validate bank account number entered by the student.

2.1.validating account number
2.1.1.validate that the account number contains 12 characters.
2.1.2.validate that the account number doesn't contain decimal numbers.
2.1.3.validate that the account doesn’t contain negative numbers
2.1.4.validate that the account number doesn’t contain any special characters in it “&*#$@’\?|<>,.”
2.1.5.validate that the account doesn't contain letters.
2.1.6.validate that the account doesn't contains any spaces.
2.2.validate bank account password that the student has entered.
2.2.1.validate that the bank account password should contains 8 characters.
2.2.2.validate that the bank account password should contains capital letters and small letters.
2.2.3.validate that the bank account password doesn’t contain any special characters in it “&*#$@’\?|<>,.”
2.2.4.validate that the bank account password doesn’t contain any spaces.2.3.The system checks that the student entered does exists.
By reaching out by the bank authorization that verifies the account.
2.4.check the bank account contains money needed that the system with draw automatically
2.4.1.validate that the money not with negatives.
2.4.2.1.Validate the calculated VAT that has been calculated by the system
2.4.2.2. Validate the calculated VAT that has been calculated by the system entered doesn't exceed 6 characters.
2.4.2.3. Validate the calculated VAT that has been calculated by the system entered is not a negative signed number.
2.4.2.4. Validate the calculated VAT that has been calculated by the system entered is a numbered sentence that doesn't contain any letters.
2.4.2.5. Validate the calculated VAT that has been calculated by the system entered should not contain any special characters in it “&*#$@’\?|<>,.”
2.4.2.6. Validate the calculated VAT that has been calculated by the system entered should only contain numbers “123456789”.
2.4.2.7. Validate the calculated VAT that has been calculated by the system entered is not >1000 EGP.
2.4.2.8. Validate the calculated VAT that has been calculated by the system entered is not <100 EGP.
2.4.3.Incase of no sufficient amount of money the system displays an

error message declined process.
3-validate master card number that the student entered.
3.1.validating the master card
3.1.1.validate that the account number contains 12 characters.
2.1.2.validate that the account number doesn't contain decimal numbers.
3.1.3.validate that the account doesn’t contain negative numbers3.1.4.validate that the account number doesn’t contain any special
characters in it “&*#$@’\?|<>,.”
3.1.5.validate that the account doesn't contain letters.
3.1.6.validate that the account doesn't contains any spaces.
3.2.validate master card expiration date entered by student.
3.2.1.validate that the master card expiration date entered by student contains 4 characters.
3.2.2.validate that the master card expiration date entered by student doesn't contain decimal numbers.
3.2.3.validate that the master card expiration date entered by student doesn’t contain negative numbers
3.2.4.validate that the master card expiration date entered by student doesn’t contain any special characters in it “&*#$@’\?|<>,.”
3.2.5.validate that the master card expiration date entered by student doesn't contain letters.
3.2.6.validate that the master card expiration date entered by student doesn't contains any spaces.
3.3.validate master card security code entered by the student
3.3.1.validate that the master card security code entered by student contains 7 characters.
3.3.2.validate that the master card security code entered by student doesn't contain decimal numbers.
3.3.3.validate that the master card security code entered by student doesn’t contain negative numbers
3.3.4.validate that the master card security code entered by student doesn’t contain any special characters in it “&*#$@’\?|<>,.”
3.3.5.validate that the master card security code entered by student doesn't contain letters.
3.3.6.validate that the master card security code entered by student doesn't contains any spaces.3.4.validate that the master card exists by reaching out by the bank
authorization that verifies the account.

3.5.check the bank account contains money needed that the system with draw automatically
3.5.1.validate that the money not with negatives.
3.5.2.1.Validate the calculated VAT that has been calculated by the system
3.5.2.2. Validate the calculated VAT that has been calculated by the system entered doesn't exceed 6 characters.
3.5.2.3. Validate the calculated VAT that has been calculated by the system entered is not a negative signed number.
3.5.2.4. Validate the calculated VAT that has been calculated by the system entered is a numbered sentence that doesn't contain any letters.
3.5.2.5. Validate the calculated VAT that has been calculated by the system entered should not contain any special characters in it “&*#$@’\?|<>,.”
3.5.2.6. Validate the calculated VAT that has been calculated by the system entered should only contain numbers “123456789”.
3.5.2.7. Validate the calculated VAT that has been calculated by the system entered is not >1000 EGP.
3.5.2.8. Validate the calculated VAT that has been calculated by the system entered is not <100 EGP.
3.5.3.In case of no sufficient amount of money the system displays an error message declined process.
4.validate the completion of money transaction that has been done by the system and student receives.
5.validate the saving of the money transaction to the database
6.Validate that all the required fields are filled with appropriate information.
6.1.Validate that username is entered.
6.1.1..Check if the username is at least 8 characters long.6.1.2.Check if the username does not contain any special character.
6.2If not, send a warning to user accordingly.
6.2.1.Validate that password is entered
6.2.2.Check if the password is at least 8 characters long.
6.2.3.Check if the password does not contain white space.
6.2.4..If not, send a warning to user accordingly.

Use case 3: Create student's semester report
Actors: Department
Description: The department user creates a report for each student each semester he/she passed. This report includes all the data for each student. Such as all the courses the students have registered in. including the course name, course code, course credit hours and the student's grade in each course. Then calculating the total credit hours for all courses , Also calculating his/her GPA and CGPA and according to his/her CGPA, If the student gets less than 2.0 CGPA the system automatically adds the student to a half load system which controls the registration of courses, If the student gets more than 2.0 CGPA the system will keep the students to his/her normal semester.

Actor intentions	System responsibility
1.The department user opens the website through the browser he chooses.	
	2.1.System gets the entered user information 2.2..System validates that such user exists. 2.3.System opens the dashboard page. 2.4.The system displays all options that are for the department only.
2.The department enters log in page info username and password	
3. The department user selects the option of creating the student's semester report.	
	4.The system displays empty report page
5. The department fills in the empty gaps in the report by entering the course code, course credit hours, course grade.	
6.The department user calculates the total number of credit hours that the student has passed in each semester. By clicking on a tab that calculates it for him and enters each course credit hour and then calculates total credit hours.	
	7.The system displays the total credit hours.
8.The department user calculates the CGPA for each semester. By clicking on a tab that calculates CGPA by entering current semester course grade and then calculates CGPA.	

10.The department user calculates the total GPA for all semesters. By clicking on a tab that calculates total GPA by entering all the semesters grades and then calculate total GPA.	9.The system starts to calculate the data that the department entered.
12.The department reviews the report and ensures that there are no errors or wrong grade	11. The system starts to calculate data that the department entered.
14.if everything is correct, the department user can save, print, or download the report.	13.If the data is correct, the system will generate semester report with the selected parameters and display it to the department user. 15. The department user returned to the system menu.

• ***Alternative courses:***Step 1: If there is no internet connection the system is responsible for
warning the user by displaying a warning message that indicates that he needs to connect to the internet through the browser he works on.

Step 5: If the department user enters invalid information, such as an incorrect course name /course code /course credit hour, the system displays an error message and prompts the department user to enter valid information before continuing.
Step 6: If the department user enters invalid information, such as an incorrect course credit hour that the student taken it or wrong course name which no one registered in from the beginning. The system displays an error message and prompts the department user to enter valid information before continuing.
Step 8/10: If the department user enters invalid information, such as an incorrect semester course grade. The system will check the grades first then display an error message and prompt the department user to enter valid information before continuing.
Step 12: If the system is unable to generate a semester report, such as due to a technical error, the system displays an error message and prompts the department user to try again later.

- **Test requirements:**

1.Validate connecting between the website system and the database management system
1.1. check if there is a wrong technical connection with database
1.1.1. If so, display a technical connection error message.
1.2. check if there is a delay in the connection due to weak connection.
1.2.1. If so, then the system display tries to connect with the database system message.
1.3. Validate to try to connect again if there is wrong with connection with database.
2.Validate the data entered by the department in the empty gaps of the report.2.1. Validating the course code entered by the department user.
2.1.1. Validate that the course code entered by the department should not contain any special characters “&*#$@’\? |<>.”
2.2.2. Validate that the course code contains three capital letters and three numbers.
2.2.3. Validate that the course code has no space between the letters and numbers.
2.2. Validate the course credit hours entered by the department user for each course.
2.2.1. The course credit hours should not contain any letters.
2.2.2. The course credit hours should only contain one number.
2.2.3. The course credit hours should only contain numbers

“01234”
2.2.4. The course credit hours should not contain any special characters in it “&*#$@’\? |<>,.”.
2.2.5. The course credit hours should not be >4 credit hours.
2.2.6. The course credit hours should not be <0 credit hours.
2.2.7. The course credit hours should not be a negative signed number.
2.2.8. The course credit hours should already be stored in the database and after the department user enters it, the database displays a successful existing message.
2.3. validate the course grade entered by the department user for each course.
2.3.1. The course grade should contain a maximum of two characters.
2.3.1.1. The course grade should contain one capital letter from the two
characters “A/B/C/D/F”.
2.3.1.2 The course grade could contain +/- from the other two characters or not.2.3.2. The course grade should not contain any numbers.
2.3.3. The course grade cannot contain decimal numbers.
2.3.4. The course grade should not contain any special characters in it “&*#$@’\? |<>,.”.
3. Validate the data entered by the department in the empty gaps of the report.
3.1 Validate the course credit hours entered by the department user for each course.
3.1.1. The course credit hours should not contain any letters.
3.1.2. The course credit hours should only contain one number.
3.1.3. The course credit hours should only contain numbers “01234”
3.1.4. The course credit hours should not contain any special characters in it “&*#$@’\? |<>,.”.
3.1.5. The course credit hours should not be >4 credit hours.
3.1.6. The course credit hours should not be <0 credit hours.
3.1.7. The course credit hours should not be a negative signed number.
3.1.8. The course credit hours should already be stored in the database and after the department user enters it, the database displays

a successful existing message.
3.2 validate the total course credit hours in each semester entered by the department user for each student.
3.2.1. The total courses credit hours should not contain any letters.
3.2.2. The total courses credit hours should only contain numbers.
3.2.3. The total courses credit hours should not contain one digit number.3.2.4. The total courses credit hours should only contain two digits numbers.
3.2.5. The total courses credit hours should not contain any special characters in it “&*#$@’\? |<>,.”.
3.2.6. The total courses credit hours should not exceed 20 credit hours in each semester.
3.2.7. The total courses credit hours should not be less than 10 credit hours in each semester.
3.2.8. The total courses credit hours should not be a negative signed number.
3.2.9. The total courses credit hours should already be stored in the database and after the department user enters it, the database displays a successful existing message.
3.3 validate the total course credit hours in all semesters entered by the department user for each student.
3.3.1. The total courses credit hours in all semesters should not contain any letters.
3.3.2. The total courses credit hours in all semesters should only contain numbers.
3.3.3. The total courses credit hours in all semesters should contain more than one digit number.
3.3.4. The total courses credit hours in all semesters could be two- or three-digits numbers.
3.3.5. The total courses credit hours in all semesters should not contain any special characters in it “&*#$@’\? |<>,.”.
3.3.6. The total courses credit hours in all semesters should not be less than 100 credit hours.
3.3.7. The total courses credit hours in all semesters should not be a negative signed number.
3.3.8 The total courses credit hours in all semesters should not contain any decimal numbers.3.3.9. The total courses credit hours in all semesters should
already be stored in the database and after the department user enters

it, the database displays a successful existing message.
4.Validate the data entered by the department in the empty gaps of the report.
4.1. validating the CGPA entered by the department user.
4.1.1. validate that the CGPA should not contain any letters.
4.1.2. validate that the CGPA can contain decimal numbers.
4.1.3. validate that the CGPA should not be a negative signed number.
4.1.4. validate that the CGPA is not > 4 GPA.
4.1.5. validate that the CGPA is not <0 GPA.
4.1.6. validate that the CGPA should not contain any special characters in it “&*#$@’\? |<>,.”.
4.2. validating the total GPA entered by the department user.
4.2.1. validate that the total GPA should not contain any letters.
4.2.2. validate that the total GPA can contain decimal numbers.
4.2.3. validate that the total GPA should not be a negative signed number.
4.2.5. validate that the total GPA is not > 4 GPA.
4.2.6. validate that the total GPA is not <0 GPA.
4.2.7. validate that the total GPA should not contain any special characters in it “&*#$@’\? |<>,.”.
5.Validate the completion of the report that has been done by system that the department user receives.
6. Validate the saving of the report to the database.7.Validate that all the required fields are filled with appropriate information.
7.1.Validate that username is entered.
7.1.1..Check if the username is at least 8 characters long.
7.1.2.Check if the username does not contain any special character.
7.2If not, send a warning to user accordingly.
7.2.1.Validate that password is entered
7.2.2.Check if the password is at least 8 characters long.
7.2.3.Check if the password does not contain white space.
7.2.4..If not, send a warning to user accordingly.

Use case 4: Add/Edit Performance
Actor: Doctor and Teaching Assistants
Description: The Doctor and IT wants to add and edit Students Performance by adding each student name and ID and course ID then add each student marks and attendance or edit a mark or attendance that already has been added then save it in system database and there is option for calculating all the grades at the end of each semester then send it to the required student as semester work

Actor intentions	System responsibility
. 1- 1-The Doctor/TA opens the website through the browser he chooses 1-2-The doctor /TA enters log in page and enter username and password	**2.1.System gets the entered user information** **2.2.System validates that such user exists.** **2.3.System opens the dashboard page.**

	2.4In case of Doctor's account The system displays all the options that are available for the Doctors only.
4-The Doctor/TA selects Edit performance option	2.5In case of TA's account The system displays all the options that are available for the TA's only
6-The Doctor/TA writes the course ID and search for it.	5-The system displays Edit performance tab
	7-The system validates the course ID to ensure that it's true and accurate. 8-The system displays the course excel sheet tab..
9-The Doctor/TA writes the required student name and ID then	8- The admin is returned to the system's administrative menu.
11-The Doctor/TA can add/edit the required student marks or attendance then saves it	10-The system validates the Student name and ID to ensure that it's true and accurate
13-The Doctor/TA can review the excel sheet then if everything is correct they can send the performance to each required student.	12-The system saves the data in system database then calculate the grades automatically for the

	total semester work.

- **Alternative courses:**

Step1: If there is no internet connection the system is responsible for warning the user by displaying a warning message that indicates that he needs to connect to the internet through the browser he works on.
Step6:If the Doctor/TA enters invalid course ID, the system displays an error message and prompts the Doctor/TA user to enter valid information before continuing..
Step9: If the Doctor/TA enters invalid student name or ID, the system displays an error message and prompts the Doctor/TA user to enter valid information before continuing.
Step11:if the Doctor/TA entered wrong data such as negative marks, the system displays an error message and prompts the Doctor/TA user to write the correct data.
Step12: if the system is unable to save the data, such as due to a technical error, the system displays an error message and prompts the Doctor/TA user to try again later.
Step13: if the system is unable to send the performance data to required student, such as due to a technical error, the system displays an error message and prompts the Doctor/TA user to try again later.

- **Test Requirements:**

1.Validate connecting between the website system and the database management system 1.1. check if there is a wrong technical connection with database
1.1.1. If so, display a technical connection error message.
1.2. check if there is a delay in the connection due to weak connection
1.2.1. If so, then the system display tries to connect with the database system message.1.3. Validate to trying to connect again if there is wrong with connection
with database 2.Validate the data entered by the Doctor/TA .
2.1. Validating the course ID entered by the Doctor/TA user
2.1.1. validate that the course ID entered should not contain any special characters “&#$@’?|<>,.
2.1.2. validate that the course ID contains capital letters and small letters.
2.1.3. The course ID should not be a negative signed number
. 2.1.4. The course ID should be already stored in the database and after

the Doctor/Ta user enters it the database displays a succeeded existing course ID message.
2.2. Validating the student name and ID entered by the Doctor/TA user.
2.2.1. validate that the name contains capital letters and small letters.
2.2.2. validate that the name doesn't contain any numbers”123456789”
2.2.3. validate that the name entered should not contain any special characters “&#$@’?|<>,.
2.2.4. validate the there is a space between the first name and the middle name and a space between the middle name and the last name.
2.2.5. validate that the ID entered should not contain any special characters “&*#$@’?|<>,. 2.2.6. validate that the ID don’t contains capital letters and small letters.
2.2.7. The course ID should not be a negative signed number.
2.2.8 validate that the ID doesn’t exceed 8 numbers
2.2.9 The student name and ID should be already stored in the database and after the Doctor/Ta user enters it the database displays a succeeded existing student message. 3.Validate the data of the performance that added by Doctor/Ta.
3.1. Validate the marks that added by The Doctor/Ta.
3.1.1. Validate that the marks is not negative integer.
3.1.2. Validate that the marks doesn’t contain characters.3.1.3 validate that the marks entered should not contain any special characters “&#$@’?|<>,.
3.1.4 validate that the marks entered doesn’t exceed 3 numbers
3.2 validate the attendance that added by the Doctor/Ta.
3.2.1 validate that the attendance is not negative integer.
3.2.2 validate that the attendance entered should not contain any special characters “&#$@’?|<>,.
3.2.3 validate that the attendance is added in the right slot.
3.2.4 validate that attendance added doesn’t exceed 1 character
3.3 validate that the data stored in the right place in the system database then enter succeeded message to the Doctor/TA.
4.Validate the data of the performance that edited by Doctor/Ta.
4.1.1 Validate that there is already data in the system database that can be edited.
4.1.2 if there is no data available to be edited display error message for the Doctor/TA.
4.2.1 Validate that the edited marks is not negative integer.
4.2.2 Validate that the edited marks doesn’t contain characters.
4.2.3 validate that the edited marks entered should not contain any

special characters “&#$@’?|<>,.
4.2.4 validate that the edited marks entered doesn’t exceed 3 numbers.
4.2.5 validate that the edited attendance is not negative integer.
4.2.6 validate that the edited attendance entered should not contain any special characters “&#$@’?|<>,.
4.2.7 validate that the edited attendance is added in the right slot.
4.2.8 validate that the edited attendance doesn’t exceed 1 character.
4.3 validate that the edited data stored in the right place in the system database then enter succeeded message to the Doctor/TA.5.validate the calculated marks for the full semester work.
5.1 Validate that the marks is not negative integer.
5.2 Validate that the marks doesn’t contain characters.
5.3 validate that the marks entered should not contain any special characters “&*#$@’?|<>,. 5.4 validate that the marks doesn’t exceed 3 numbers
6.validate that the system saved the data in the system database
7.Validate that all the required fields are filled with appropriate information.
7.1.Validate that username is entered.
7.1.1..Check if the username is at least 8 characters long.
7.1.2.Check if the username does not contain any special character.
7.2If not, send a warning to user accordingly.
7.2.1.Validate that password is entered
7.2.2.Check if the password is at least 8 characters long.
7.2.3.Check if the password does not contain white space.
7.2.4..If not, send a warning to user accordingly.
8.validate that the Doctor/TA sent the performance that sent to the specific student
8.1 if there is an issue with the sent data the system displays error message to the Doctor/TA.

Use case5: Add new user.
Actor: Department
Description: This use case allows the department to add new users to the system by adding their information such as name, password ,email

and id ,The system then verifies the information provided and creates a new user account. Once the account is created, the user is notified and can log in to access the platform's functionalities.

Actor intentions	System responsibility
1-1-The department user opens the website through the browser he chooses. 1-2-The department user enters log in page and enters username and password	**2.1.System gets the entered user information** **2.2System validates that such user exists.** **2.3.System opens the dashboard page.**
3-The department user selects the options of adding new user such as DR/TA/ Student	2.4.The system displays all options that are for the department only.
5-. The department fills out the form with the new user's information.	4- The system presents a form for the admin to enter the new user's information, such as their name, email address, and password.

7- If the information is valid, the system creates a new user account with the information provided and sends a confirmation email to the new user	6- The system validates the information to ensure that it is complete and accurate. 8- The admin is returned to the system's administrative menu.

- **Alternative courses:**

Step1: If there is no internet connection the system is responsible for warning the user by displaying a warning message that indicates that he needs to connect to the internet through the browser he works on.
Step 4: If the department user enters invalid information, such as an incorrect employee name or employee ID, the system displays an error message and prompts the department user to enter valid information before adding the new user .Step 5: If the department user enters invalid information, such as an
Exist user , the system displays an error message and prompts the department user to enter valid information
Step 6: if the system found that the information is uncomplete and

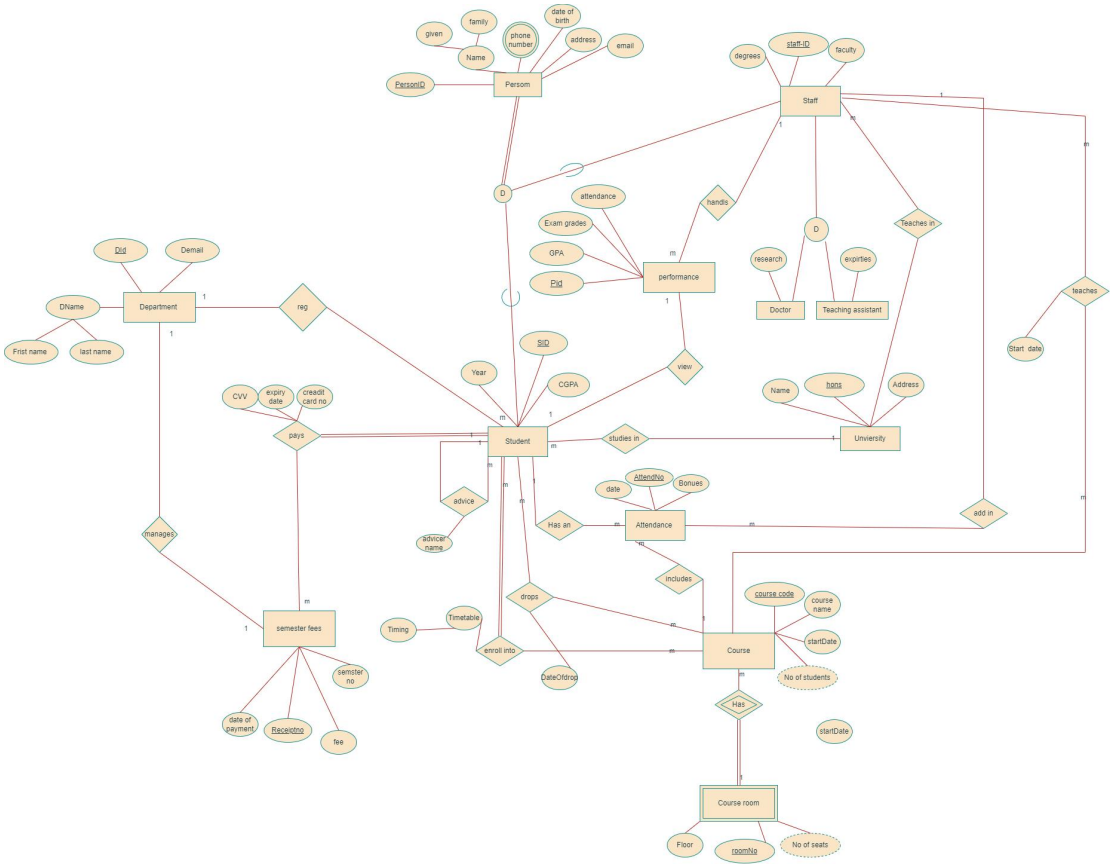
inaccurate the system displays an error message and prompts the department user to enter complete and accurate information
Step 7: If the information is valid, and the system can not create a new user, the system displays an error message to try again later.

- **Test Requirements:**

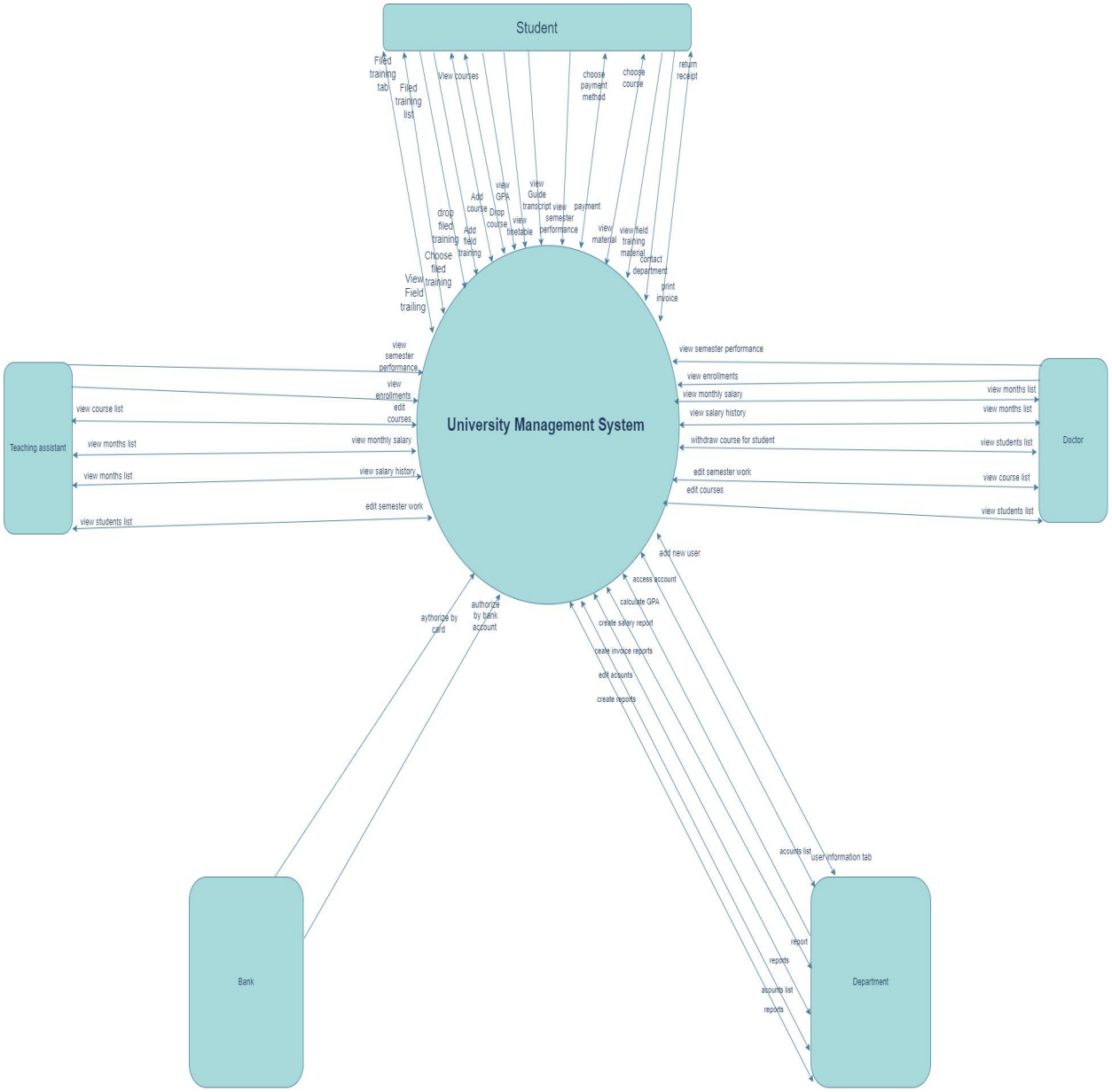
1.Validate connecting between the website system and the database management system
1.1. check if there is a wrong technical connection with database
1.1.1. If so, display a technical connection error message.
1.2. check if there is a delay in the connection due to weak connection
1.2.1. If so, then the system display tries to connect with the database system message.
1.3. Validate to trying to connect again if there is wrong with connection with database
2.validate that the department must select Student or Doctor or TA to create account
2.1 if nothing is selected the system displays error message
3.Validate the information that entered by department user
3.1.1 Validating the username entered by the department user
3.1.2. validate that the name entered should not contain any special characters &#$@'?|<>,.
3.1.3 validate that the name contains capital letters and small letters.
3.1.4 validate that the name doesn't contain any numbers"123456789"3.1.5 validate the there is a space between the first name and the
middle name and a space between the middle name and the last name.
3.2Validating the user ID entered by the department user
3.2.1 The user ID should not contain any letters
3.2.2 validate that there is no existing ID with the same number
3.2.3 The user ID should only contain numbers "123456789"
3.2.4 The user ID should be a numbered sentence with is 8 characters
3.2.5 The user ID should not contain any special characters in it "&#$@'?|<>,.".
3.2.6 The user ID should not be a negative signed number.
3.3 validate the password entered by department user
3.3.1 Validate that the user password is not less that 8 characters
3.3.2 validate that the password contains capital and small letters.
3.3.3 validate that the password contains any 2 characters in it "&#$@'?|<>,.".
3.4 validate that the university email is available and not already used

for another user
3.4.1 if its not available account or not existed the system displays error message for the department user
3.5 validate the user phone number.
3.5.1 The user phone number should not contain any letters
3.5.2 validate that there is no existing phone number with the same number
3.5.3 The user phone number should only contain numbers "123456789"
3.5.4 The user phone number shouldn't be less than 12 number
3.5.5 The user phone number should not contain any special characters in it "&#$@'?|<>,.".
3.5.6 The user ID should not be a negative signed number.4.validate that the all information is correct and filled then display
create account button
4.1 if there is any issue in the system display error message
5. Validate the completion of the user account then the system saves it in the system database.
6.Validate the saving of the user account in the system database
6.1 if there is any issue in the saving process display error message for the department user to try again
7.Validate that all the required fields are filled with appropriate information.
7.1.Validate that username is entered.
7.1.1..Check if the username is at least 8 characters long.
7.1.2.Check if the username does not contain any special character.
7.2If not, send a warning to user accordingly.
7.2.1.Validate that password is entered
7.2.2.Check if the password is at least 8 characters long.
7.2.3.Check if the password does not contain white space.
7.2.4..If not, send a warning to user accordingly.

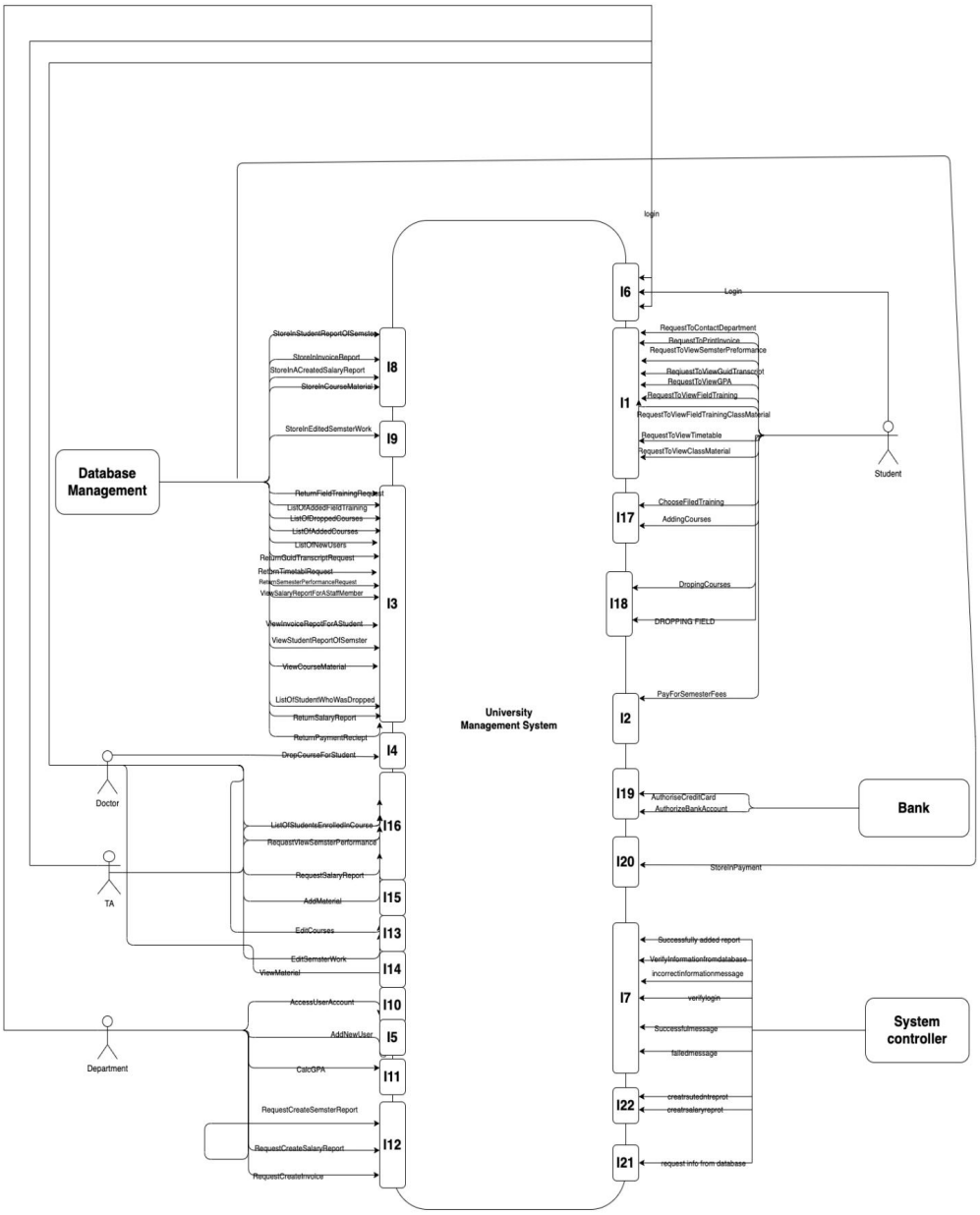
ERD:

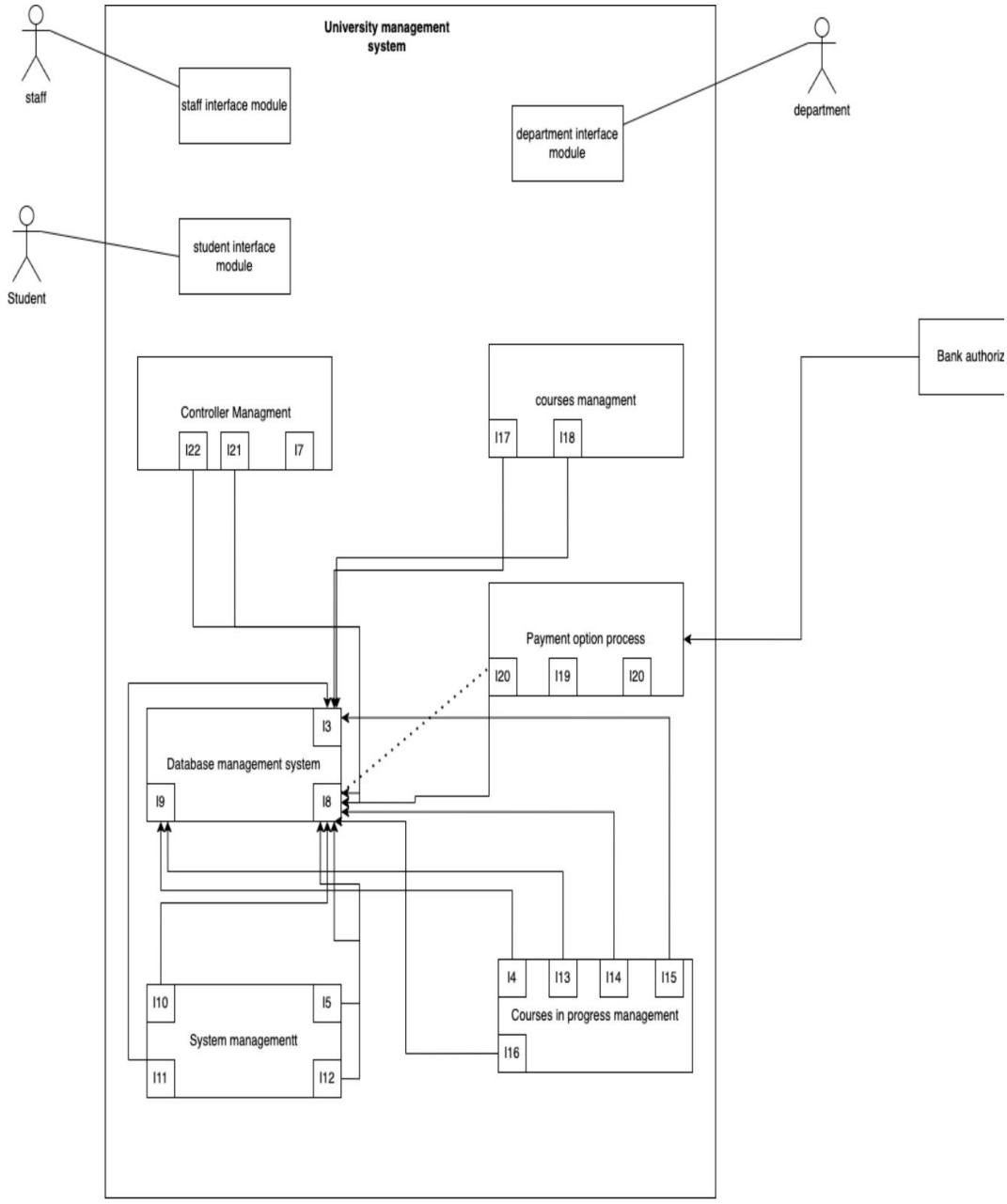

given
family
Name
phone number
date of birth
address
email
PersonID
Persom
degrees
staff-ID
faculty
Staff
D
attendance
Exam grades
GPA
Pid
performance
handls
research
expirties
Doctor
Teaching assistant
Teaches in
teaches
Start date
Did
Demail
DName
Department
reg
First name
last name
SID
Year
CGPA
view
Name
hons
Address
CVV
expiry date
creadit card no
pays
Student
studies in
University
advice
adviser name
date
AttendNo
Bonues
Has an
Attendance
add in
manages
includes
drops
Timing
Timetable
enroll into
course code
course name
startDate
Course
No of students
DateOfdrop
Has
startDate
semester fees
semster no
date of payment
Receiptno
fee
Course room
Floor
roomNo
No of seats

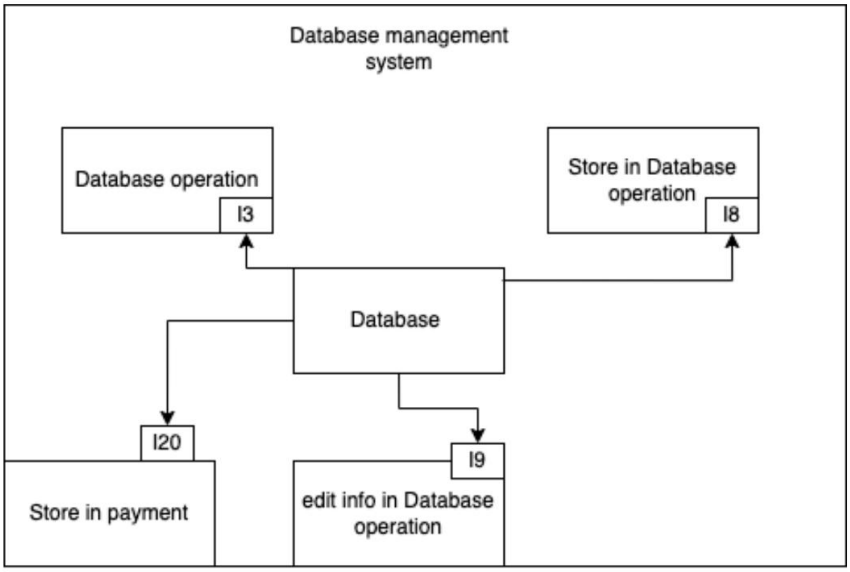
2.Context diagram :

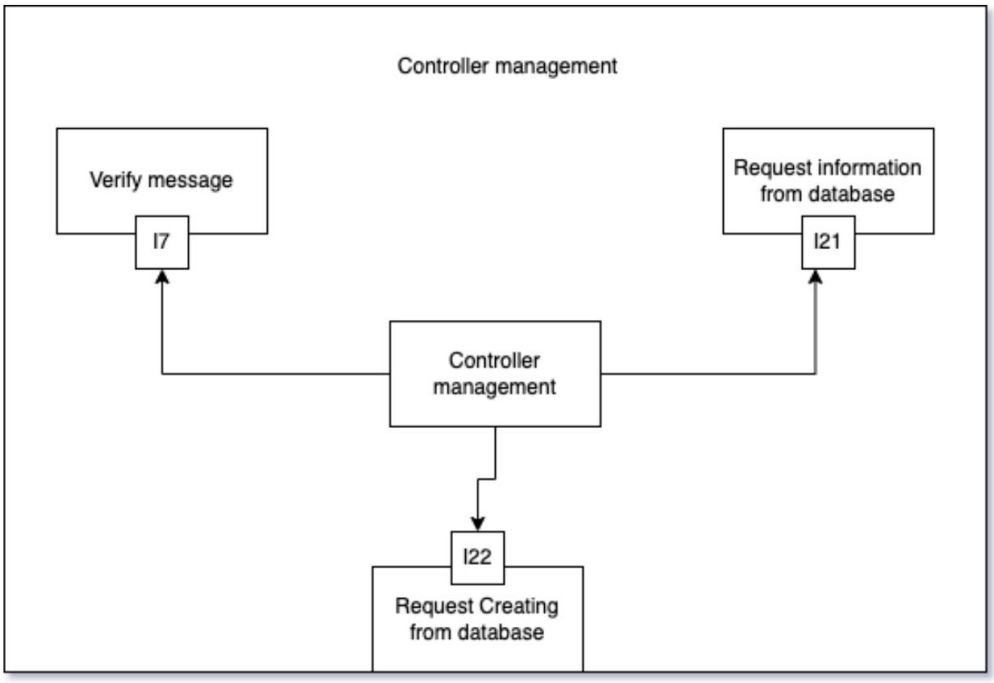


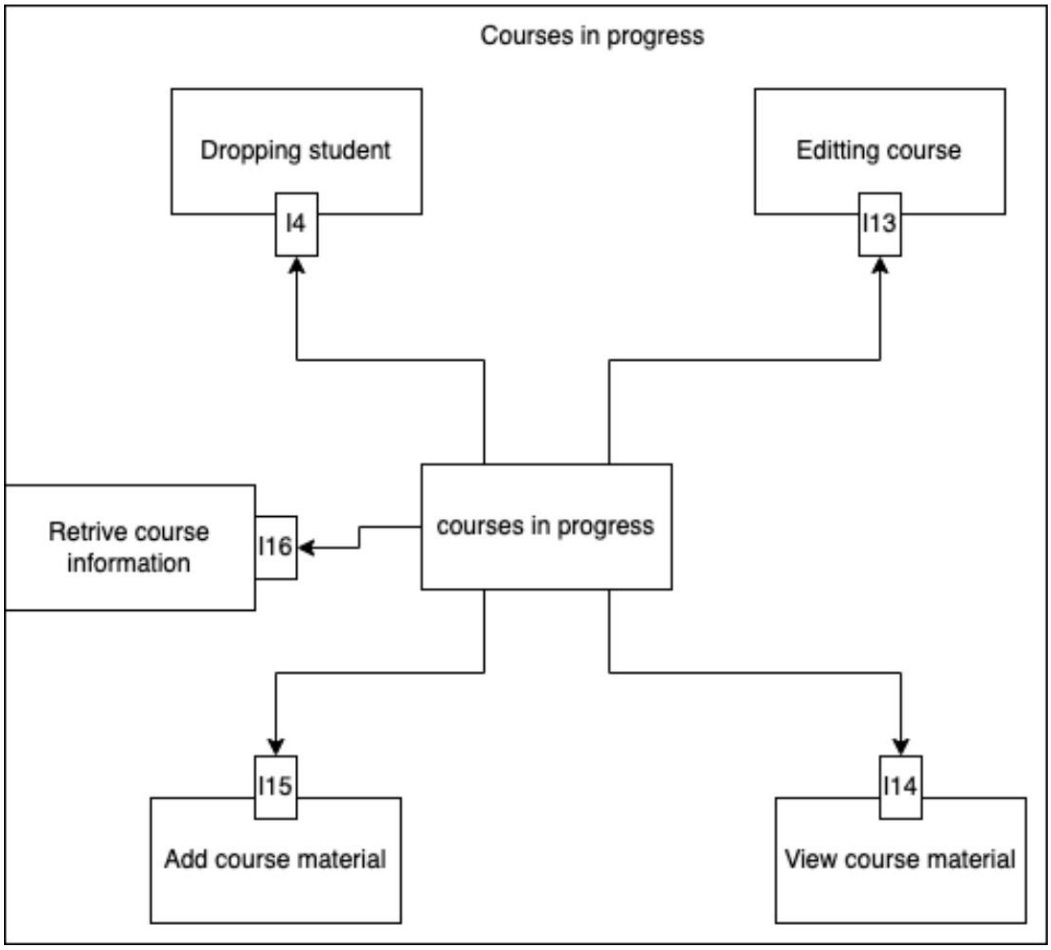
2.1. The architecture models:

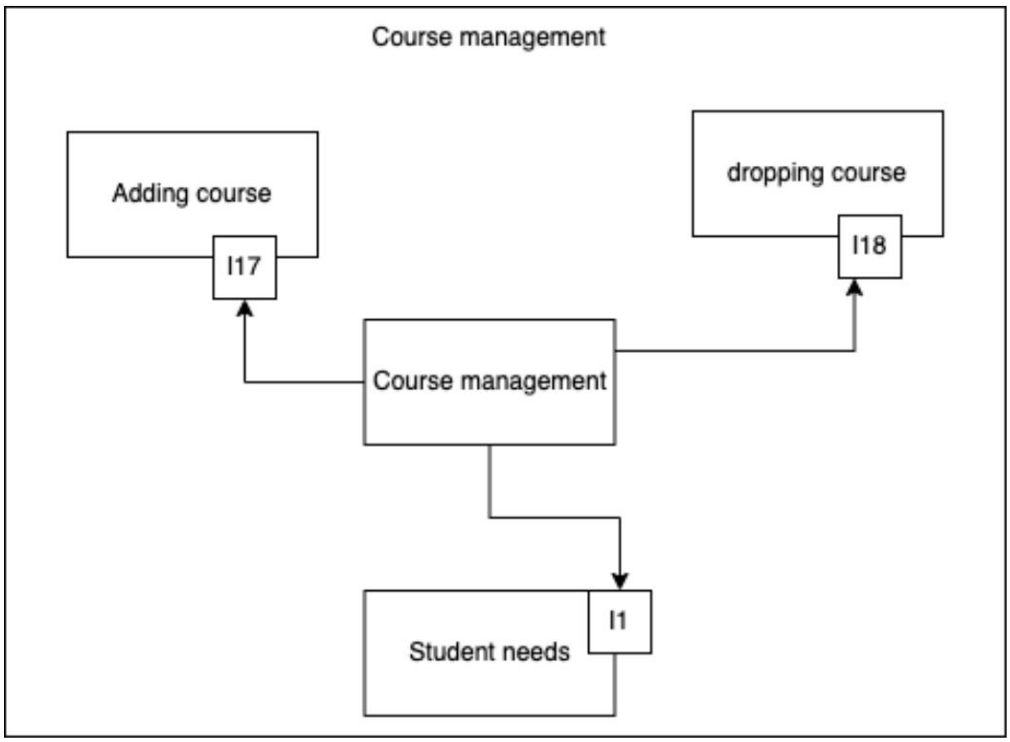


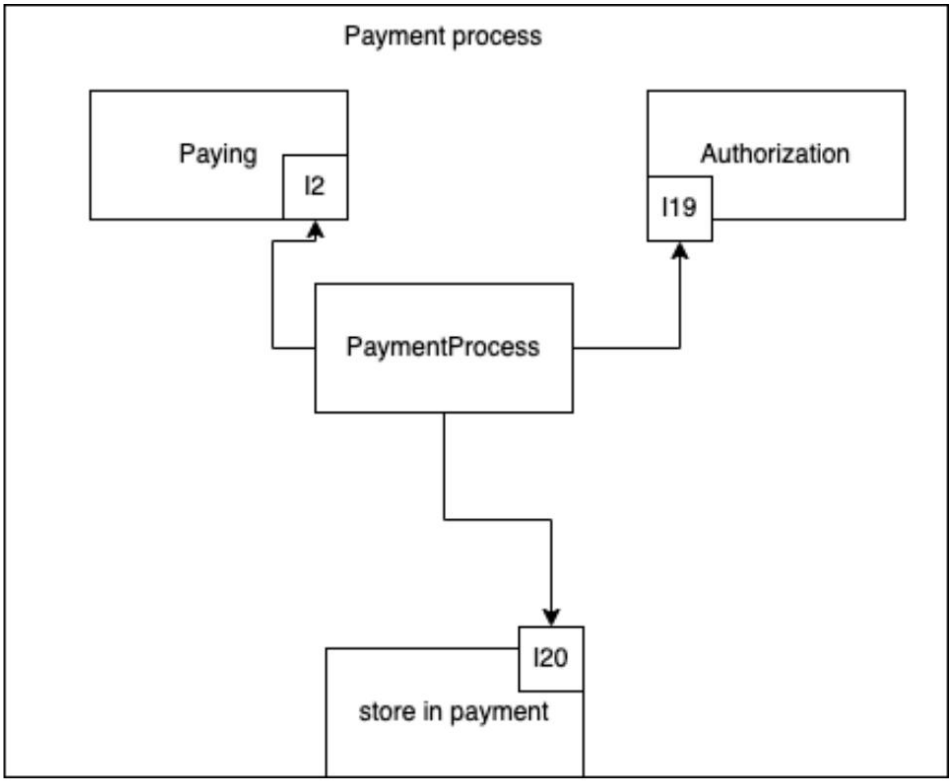

University management system
staff
staff interface module
department
department interface module
Student
student interface module
Bank authoriz
Controller Managment
I22
I21
I7
courses managment
I17
I18
Payment option process
I20
I19
I20
I3
Database management system
I9
I8
I10
I5
System managementt
I11
I12
I4
I13
I14
I15
Courses in progress management
I16

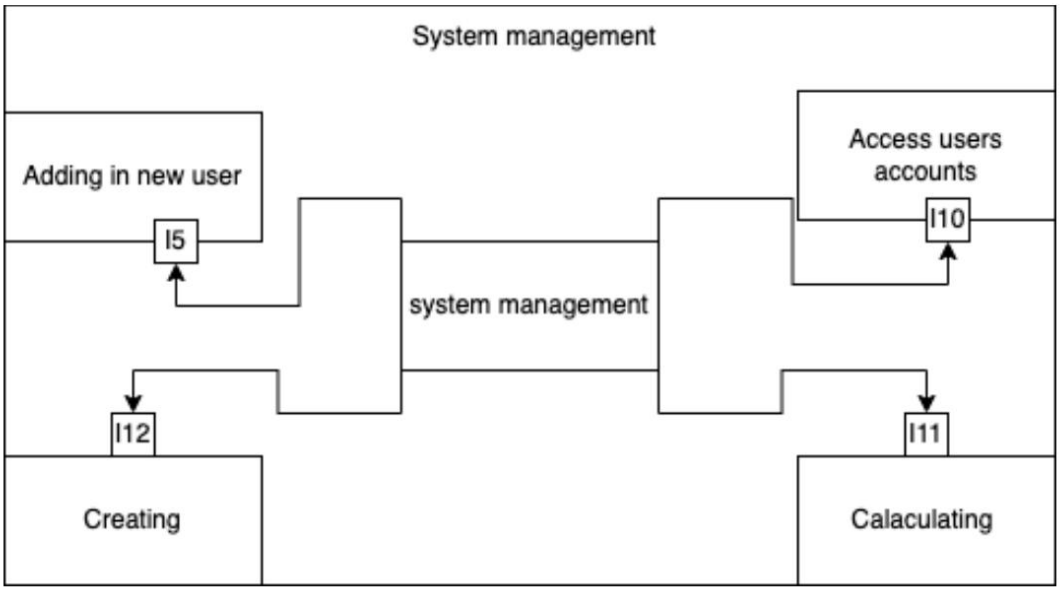

Database management system
Database operation
I3
Store in Database operation
I8
Database
I20
Store in payment
I9
edit info in Database operation


Controller management
Verify message
I7
Request information from database
I21
Controller management
I22
Request Creating from database


Courses in progress
Dropping student
I4
Editting course
I13
Retrive course information
I16
courses in progress
I15
Add course material
I14
View course material


Course management
Adding course
I17
dropping course
I18
Course management
Student needs
I1





3. Interface Definitions:

Interface Definitions:

Interface I1: Students operations interface

• Public void requesttoContactDepartment (String request);
• Public void requesttoprintinvoice (String invoicenumber);
• Public void requesttoviewsemesterperformance (String currentsemester);
• Public void requesttoviewguidetranscript (String semester);
• Public void requesttoviewGPA ();
• Public void requesttoviewfieldtraining (List<field training > filed);
• Public void requesttoviewtimetable (List<courses > course);
• Public void requesttoviewclassmaterials (List<courses > course);
• Public void requesttoviewfieldtrainingclassmaterials (List<field training >
filed);

Interface I17: Adding Courses Interface

• Public void choosesfieldtraining (List<field training > filed);
• Public void addcourses (List<courses > course);

Interface I18:Dropping Courses Interface

• Public void dropcourses (List<courses > course);
• Public void dropfieldtraining (List<field training > filed);

Interface I2: Payment interface

• Public void payforsemesterfeesbybankaccount(int accountnumber);
• Public void payforsemesterfeesbycreditcard(int creditnumber,int cvv);

Interface I19: Authorize Interface

• Public void authorizesbankaccount(int accountnumber);
• Public void authorizesbycreditcard(int creditnumber,int cvv);

Interface I20: StoreIn Payment Interface

• Public void storeinpayment(double value, int accountnumber);• Public void storeinpayment(double value, int creditnumber,int cvv);

Interface I3: Database operation interface

• Public void returnfieldtraininglist(List<field training > filed);
• Public void returnfieldtrainingmaterial(List<field training > filed);
• Public void returnfieldtrainingaddedlist(List<field training > filed);
• Public void returnfieldtrainingdropedlist(List<field training > filed);
• Public void returncourseslist(List<courses> courses);
• Public void returncoursesmaterial(List<courses > courses);
• Public void returncoursesaddedlist(List<courses > courses);

• Public void returncoursesdropedlist(List<courses > courses);
• Public void returnlistofnewusers(List<user>user);
• Public void returnguidetranscriptrequest(String semester);
• Public void returntimetablerequest(String semester, Student stu);
• Public void returnsemesterperformancerequest(String semester,Student
stu);
• Public void viewsalaryreportfordoctor(Doctor doc, Salaryreport report);
• Public void viewsalaryreportforTA(Teachingassistant ta, Salaryreport report);
• Public void viewinvoicereportforstudent(Student stu ,String semester);
• Public semseterreport(Student stu);
• Public void listofstudentswhowasdroped(List<student>stu, Course course);
• Public void returnsalaryreportfordoctor(Doctor doc, Salaryreport report);
• Public void returnsalaryreportforTA(Teachingassistant ta, Salaryreport report);
• Public void returnpaymentreciept(Student stu,String semester,Invoicereport report);
• Public void verifylogin(String username, String Password);• Public void statues ();

Interface 18:Store In DataBase

• Public void storeininvoicereport(Student stu ,String semester, Invoicereport
report);
• Public void storeincoursecourselecture(String lecture, Course course);
• Public void storeincoursecoursetutorial(String tutorial, Course course);
• Public void storeincoursecourserecordings(String recordings, Course course);
• Public void storeineditedsemesterattendance(Student stu, Course course,int attendance);
• Public void storeineditedsemestermarks(Student stu, Course course,double
marks);
• Public void storeincreatedsalaryreport(Salaryreport report);

Interface 19: Edit Information

- Public void StoreInEdit semester Mark(Student stu,Course course,StringMarks);

- Public void StoreInEdit semester Mark(Student stu,Course course,StringAttendance);
- Public void StoreInDropStudents(Student stu,Course course);

Interface I4: Dropping Students interface

• Public void dropastudent(Student stu, Course course);

Interface I13: Editing Course Interface

• Public void editcoursesattendance(int attendance, Course course,Student stu);
• Public void editcoursesmarks(int attendance, Course course,double marks,Student stu);

Interface I14: View Course Material Interface

• Public void viewlectureofcourse(String lecture, Course course);
• Public void viewrecordingsofcourse(String recordings, Course course);
• Public void viewtutorialsofcourse(String tutorials, Course course);

Interface I15: Add Class Material Interface

• Public void addlectureofcourse(String lecture, Course course);
• Public void addrecordingsofcourse(String recordings, Course course);
• Public void addtutorialsofcourse(String tutorials, Course course);

Interface I16: Retrive course Information Interface

• Public void listofstudentsenrolledincourse(List<Students>stu,Course course);
• Public void requestviewsemesterperformance(Student stu,Course course,String semester);
• Public void requestsalaryreportfordoctor(Doctor doc, String month);
• Public void requestsalaryreportforTA(Teachingassistance ta, String month);

Interface I5: Department operations interface

• Public void addnewdoctor(Doctor doc);
• Public void addnewteachingassistant(Teachingassistant ta);
• Public void addnewstudent(Student stu);

Interface I10: Accessing Accounts Interface

• Public void accessstudentaccount(Student stu);• Public void accessdoctoraccount(Doctor doc);
• Public void accessstudentaccount(Student stu);
• Public void accessteachingassistanceaccount(Teachingassistance ta);

Interface I11: Calculating Interface

• Public void calculateGPA(double CGPA,List<Course>course,Student stu,String semester);
• Public void calculateCGPA(double oldCGPA,double GPA);

Interface I12:Creating Interface

• Public void requestcreatesalaryreportfordoctor(double salary,Doctor doc,String month, double VAT);
• Public void requestcreatesalaryreportforTA(double salary,Teachingassistant ta, String month, double VAT);
• Public void requesttocreateinvoice(Student stu, double amount, double VAT, String semester);
• Public void requestcreatesemesterreport(double GPA, double CGPA,,List<Course>course,Student stu,String semester);

Interface I6: Login interface

• Public void EnterLogin(String Username, String Password);
•

Interface I7: Controller Management Interface

- Public void succefulcardinfo();
- Public void faildprocess();
- Public void succefulbankaccinfo();
- Public void verifyeditperformancerequest(Doc doctor,TeachingAssistent Acc);
- Public void displayeditperformanceoptions();
- Public void verifyeditcoursemark(Attendence att,Course cou,String mark,Student stu);
- Public void succefullyadded();
- Public void unsuccefullywronginfoentered();
- Public void verifyeditcourseAttendence(Attendence att,Course cou,String mark,Student stu);
- Public void verifyaddednewuserequest();
- Public void returnListOfUsers(List<User> user);
- Public void verifynewuserinfo(User user);

• Public void verifyuserinformationfromdatabase(String username, String password);
• Public void loginsuccssefully();
• Public void loginfailed();
• Public void verifysalaryrecordenteredinfo(double salary, String

month,doubleVAT);
• Public void informationnotcorrect();
• Public void successfullyaddedreportbydoctor(Salaryreport salaryreport);
• Public void successfullyaddedreportbyTA(Salaryreport salaryreport);
• Public void verifyreportenteredinformation(Student stu,double GPA,Double
CGPA,Course course,String semester);
• Public void informationincorrect():
• Public void unsuccessfullyaddedreportforstudent(Semesterreport report);

Interface I21: Requesting Information From DataBase Interface

● Public void requestpaymentforsemesterdetails(Student stu,Invoice invoice);● Public void storecreditcardinfo(int creditno,int cvv);
• Public void requestinvoicePrint(String invoicenumber);
• Public void requestsemisterperformance(String current semester);
• Public void requestguidtranscript(String semester);
• Public void requestGPA(Student stu);
• Public void requestfieldtraining();
• Public void requestchoosefieldtraining(Fieldtrainig field);
• Public void requestdropfieldtraining (Fieldtraining field);
• Public void requestaddcourse(Course course);
• Public void requestdropcourse(Course course);
• Public void requestTimetable(Student stu, String semester);
• Public void requestClassMaterial(Course course);
• Public void requestFieldtrainigclassmaterial(Fieldtraining field);
• Public void requesttopayforsemesterfeesbybank(int accountnumber);
• Public void requesttopayforsemesterfeesbycreditcard(int creditnumber,int
cvv);

Interface I22: Request Creating Interface

• Public void createsalaryreportfordoctor(double salary,Doctor doctor, String
month, double VAT);
• Public void createsalaryreportforTA(double salary, Teachingassistant TA,
String month, double VAT);

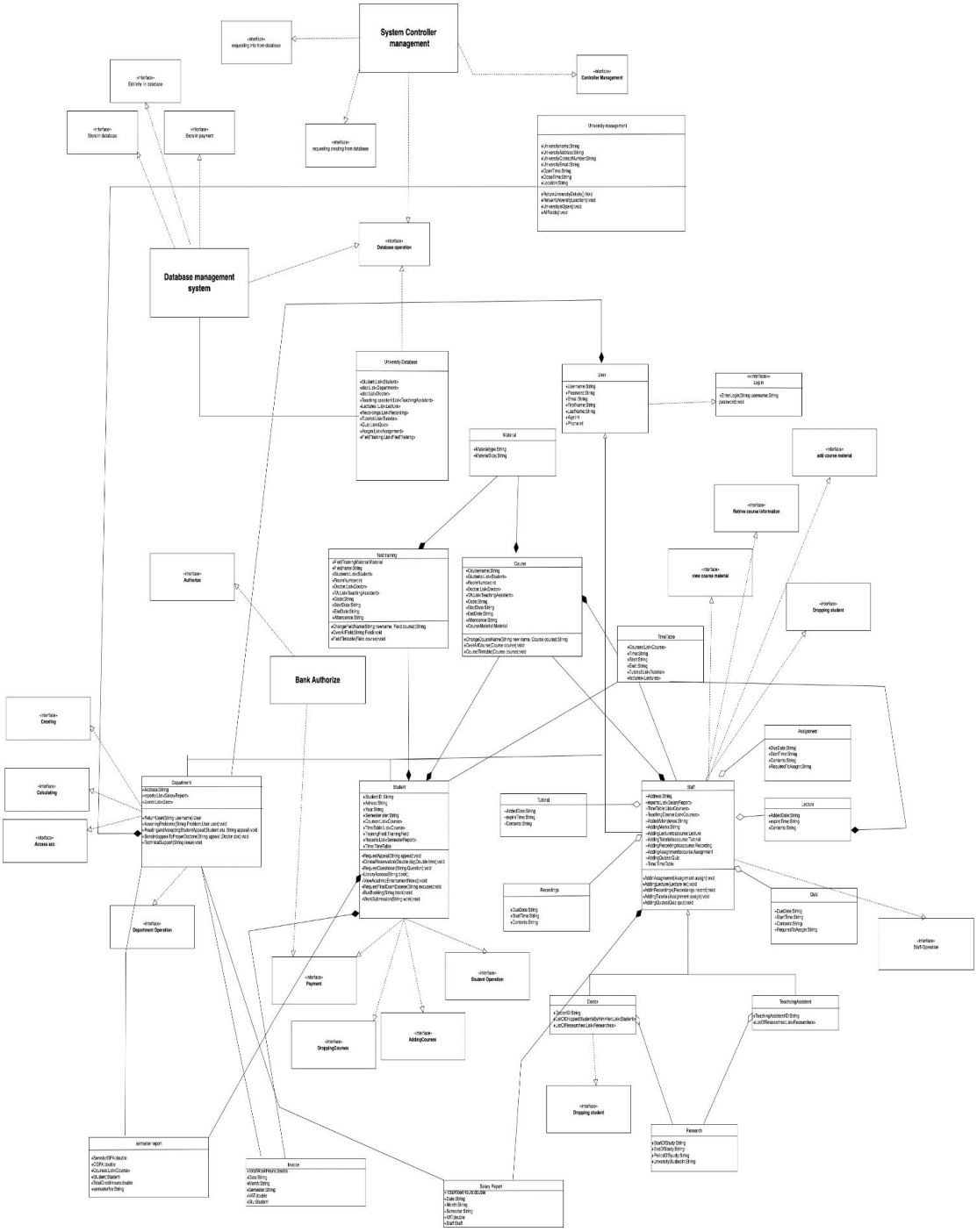
Interface 16: Login interface

• Public void EnterLogin(String Username, String Password);

Interface 17: Controller Management Interface

• Public void requestinvoicePrint(String invoicenumber);• Public void requestsemisterperformance(String current semester);
• Public void requestguidtranscript(String semester);
• Public void requestGPA(Student stu);
• Public void requestfieldtraining();
• Public void requestchoosefieldtraining(Fieldtrainig field);
• Public void requestdropfieldtraining (Fieldtraining field);
• Public void requestaddcourse(Course course);
• Public void requestdropcourse(Course course);
• Public void requestTimetable(Student stu, String semester);
• Public void requestClassMaterial(Course course);
• Public void requestFieldtrainigclassmaterial(Fieldtraining field);
• Public void requesttopayforsemesterfeesbybank(int accountnumber);
• Public void requesttopayforsemesterfeesbycreditcard(int creditnumber,int
cvv);
• Public void createsalaryreportfordoctor(double salary,Doctor doctor, String
month, double VAT);
• Public void createsalaryreportforTA(double salary, Teachingassistant TA,
String month, double VAT);

4. Class Diagram:


System Controller management
«Interface» requesting info from database
«Interface» Controller Management
«Interface» Edit info in database
«Interface» Store in database
«Interface» Store in payment
«Interface» requesting creating from database
University management
Database management system
«Interface» Database operation
University Database
User
«Interface» Log in
Material
«Interface» add course material
«Interface» Retrive course information
«Interface» Authorize
Bank Authorize
Field training
Course
«Interface» view course material
«Interface» Dropping student
TimeTable
«Interface» Creating
«Interface» Calculating
«Interface» Access acc
Department
Student
Tutorial
Staff
Assignment
Lecture
Recordings
Quiz
«Interface» Department Operation
«Interface» Payment
«Interface» Student Operation
«Interface» Staff Operation
«Interface» DroppingCourses
«Interface» AddingCourses
Doctor
TeachingAssistant
«Interface» Dropping student
Research
semester report
Invoice
Salary Report

Use case name	University management	Database management	user	course	staff	assignment	lecture	quiz
View field training material list		✓						
Choose field training		✓						
Drop courses		✓		✓				
Add course		✓		✓				
View GPA		✓						
View timetable		✓		✓	✓		✓	
View course martial		✓		✓	✓			
View guide transcript		✓		✓				
View semester performance		✓				✓		✓
Print invoice		✓			✓			
View field training list		✓						

Contact department					✓			
Add course in student timetable		✓		✓				
Payment		✓			✓			
View enrollments		✓		✓				
Edit course		✓		✓			✓	
Edit semester work		✓			✓			
Withdraw course for students		✓		✓	✓			
Edit attendance		✓			✓			
Edit marks		✓			✓			
View salary					✓			
View salary history		✓			✓			
View monthly salary		✓			✓			
Drop field training		✓						

Add field training		✓						
Pay by master card	✓							
Pay by bank	✓							
Authorize by card	✓							
Authorize by bank account	✓							
Add/view material		✓			✓			
Add lecture		✓					✓	
Add recordings		✓			✓			
Add tutorial		✓			✓			
Add new user information		✓	✓					
Add new user		✓	✓					
Edit accounts		✓	✓					
Access accounts		✓	✓					

Calculate GPA		✓						
Create salary report		✓						
Send salary								
Create invoice report		✓						
Send invoice								
Create reports		✓						
Send reports								

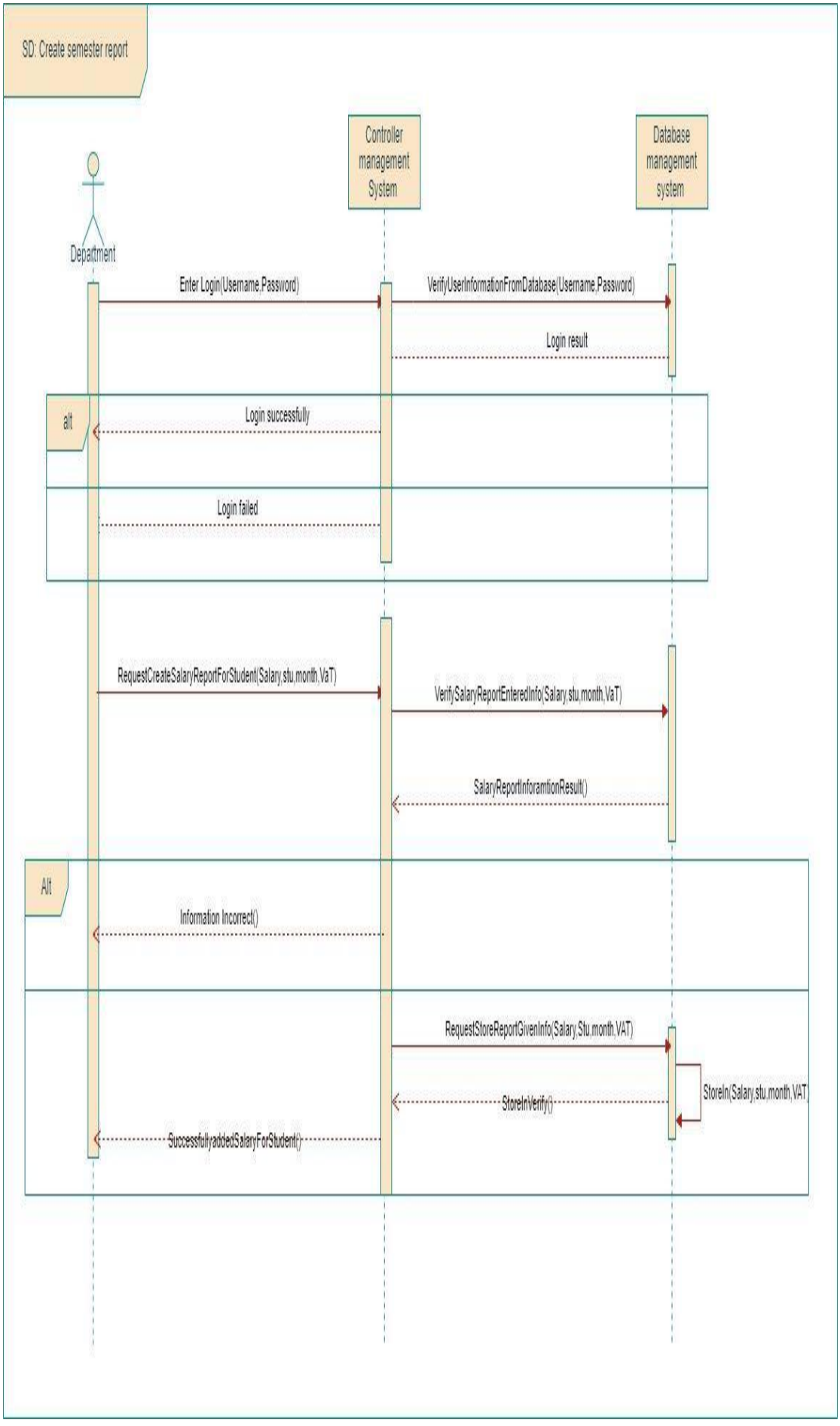
Use case name	Field training	Material	Time table	TA	Doctor	student	Department	Semester report	salary	invoice	tutorial	Recordings
View field training material list	✓					✓						
Choose field training	✓					✓						
Drop courses						✓						
Add course						✓						

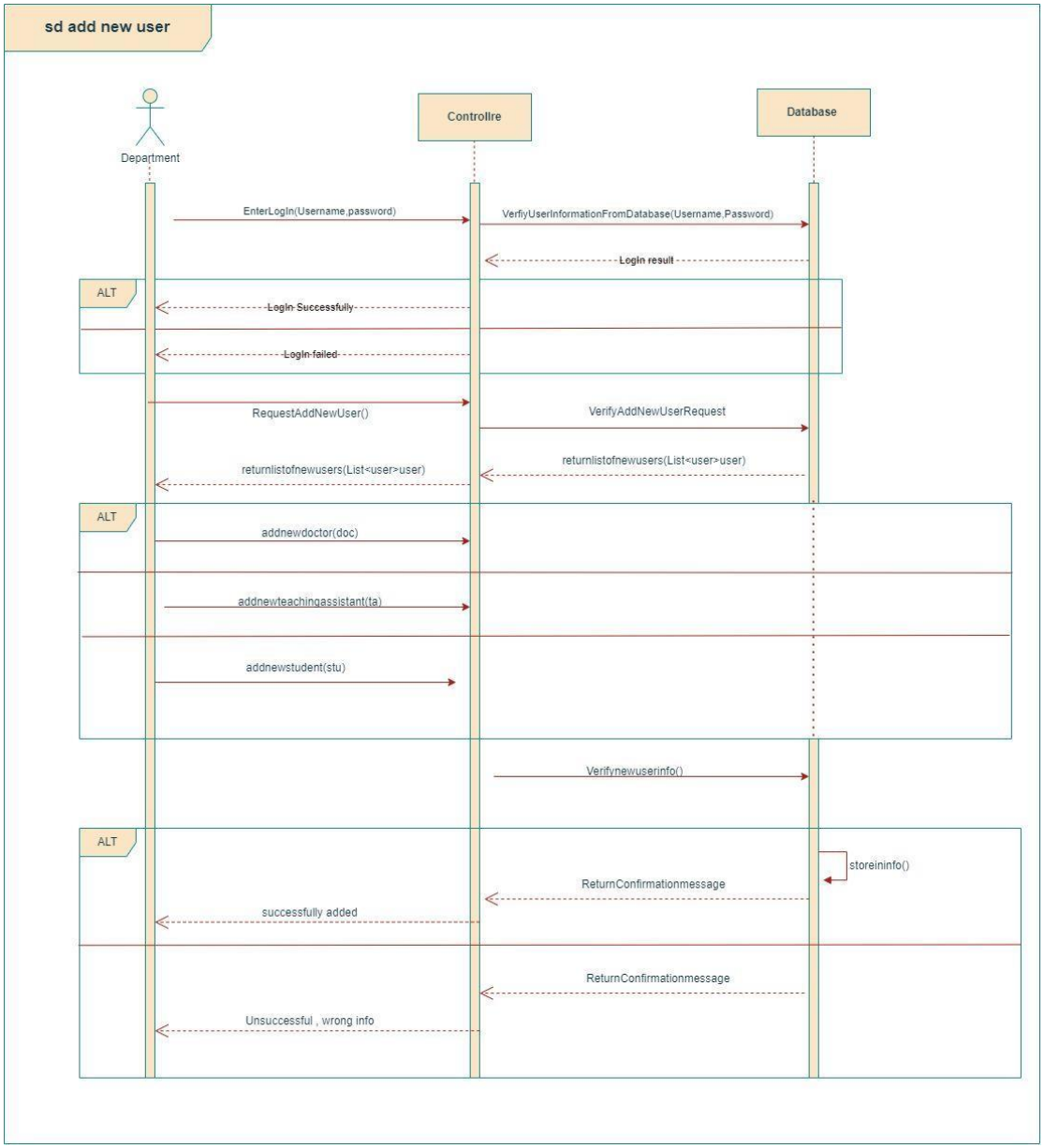
View GPA						✓						
View timetable						✓						
View course martial		✓				✓						
View guide transcript						✓						
View semester performance						✓		✓				
Print invoice				✓	✓					✓		
View field training list	✓					✓						
Contact department						✓	✓					
Add course in student timetable			✓			✓						
Payment						✓				✓		
View enrollments						✓						
Edit course				✓	✓							

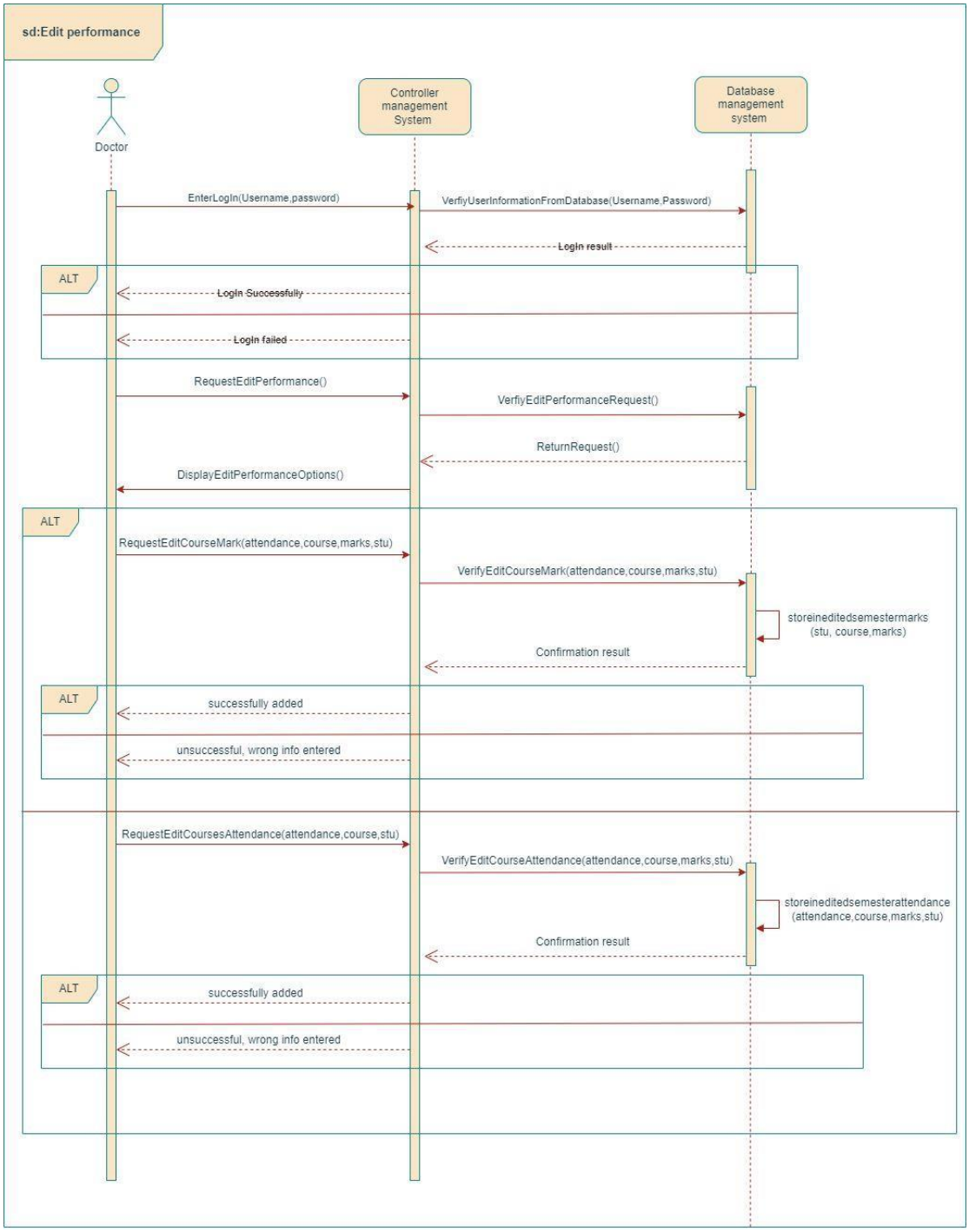
Edit semester work				✓	✓			✓				
Withdraw course for students					✓							
Edit attendance				✓	✓							
Edit marks				✓	✓							
View salary				✓	✓				✓			
View salary history				✓	✓				✓			
View monthly salary				✓	✓				✓			
Drop field training	✓					✓						
Add field training	✓					✓						
Pay by master card						✓				✓		
Pay by bank						✓				✓		
Authorize by card										✓		
Authorize by bank account										✓		

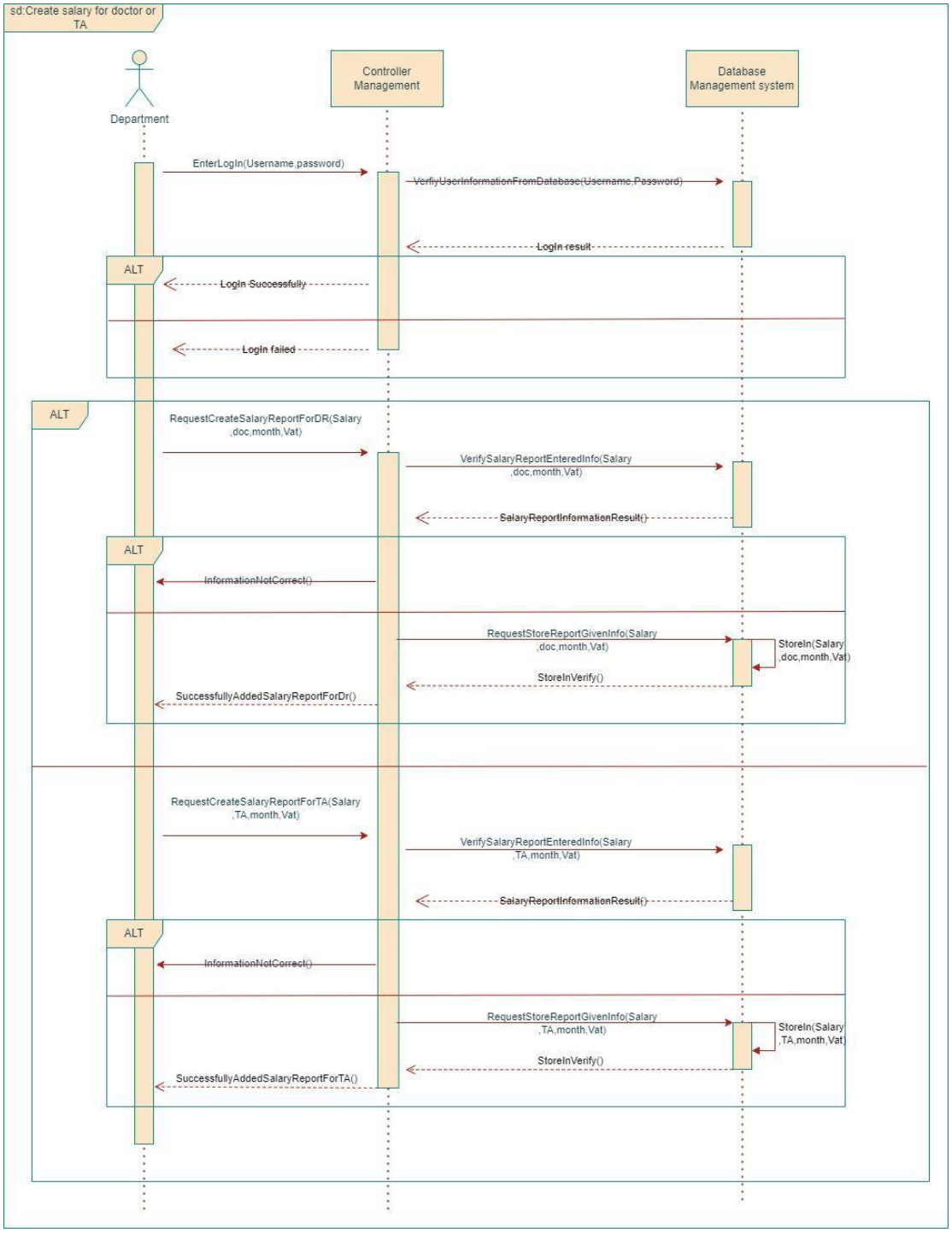
Add/view material		✓		✓	✓							
Add lecture				✓	✓							
Add recordings				✓	✓							✓
Add tutorial				✓							✓	
Add new user information							✓					
Add new user							✓					
Edit accounts							✓					
Access accounts							✓					
Calculate GPA							✓	✓				
Create salary report							✓		✓			
Send salary							✓		✓			
Create invoice report							✓			✓		
Send invoice							✓			✓		
Create reports							✓	✓				
Send reports							✓	✓				

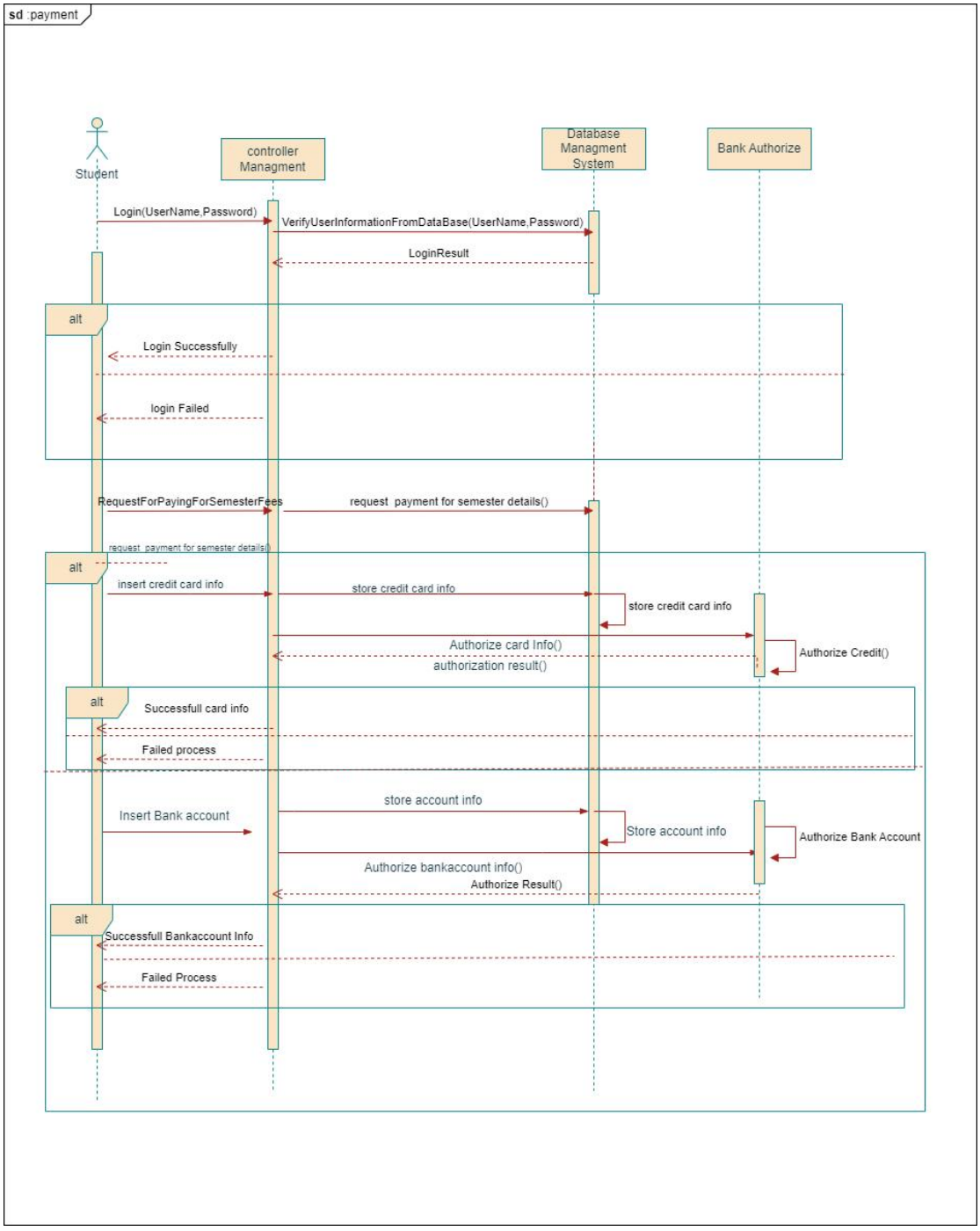
6. The sequence diagrams


SD: Create semester report
Controller management System
Database management system
Department
Enter Login(Username,Password)
VerifyUserInformationFromDatabase(Username,Password)
Login result
alt
Login successfully
Login failed
RequestCreateSalaryReportForStudent(Salary,stu,month,VaT)
VerifySalaryReportEnteredInfo(Salary,stu,month,VaT)
SalaryReportInforamtionResult()
Alt
Information Incorrect()
RequestStoreReportGivenInfo(Salary,Stu,month,VAT)
StoreIn(Salary,stu,month,VAT)
StoreInVerify()
SuccessfullyaddedSalaryForStudent()

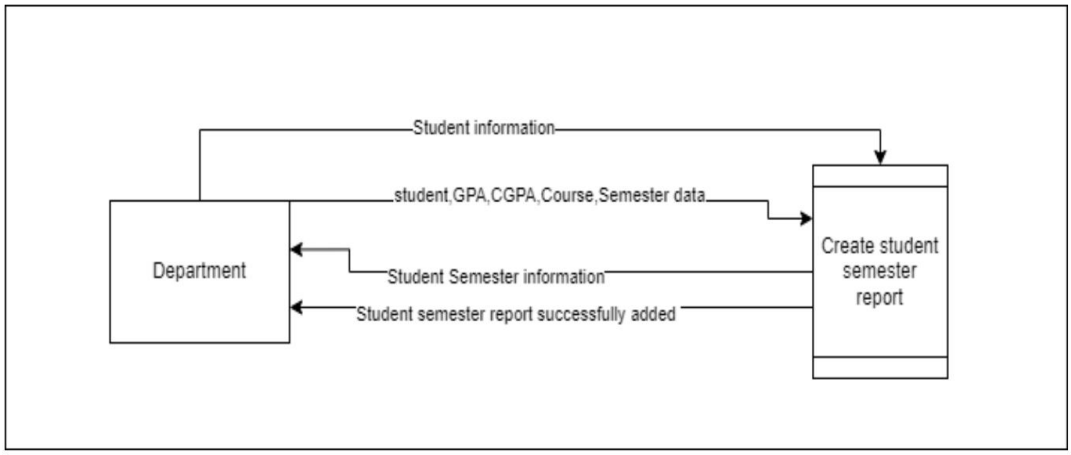

sd add new user
Department
Controllre
Database
EnterLogIn(Username,password)
VerfiyUserInformationFromDatabase(Username,Password)
Login result
ALT
Login Successfully
Login failed
RequestAddNewUser()
VerifyAddNewUserRequest
returnlistofnewusers(List<user>user)
returnlistofnewusers(List<user>user)
ALT
addnewdoctor(doc)
addnewteachingassistant(ta)
addnewstudent(stu)
Verifynewuserinfo()
ALT
storeininfo()
ReturnConfirmationmessage
successfully added
ReturnConfirmationmessage
Unsuccessful , wrong info

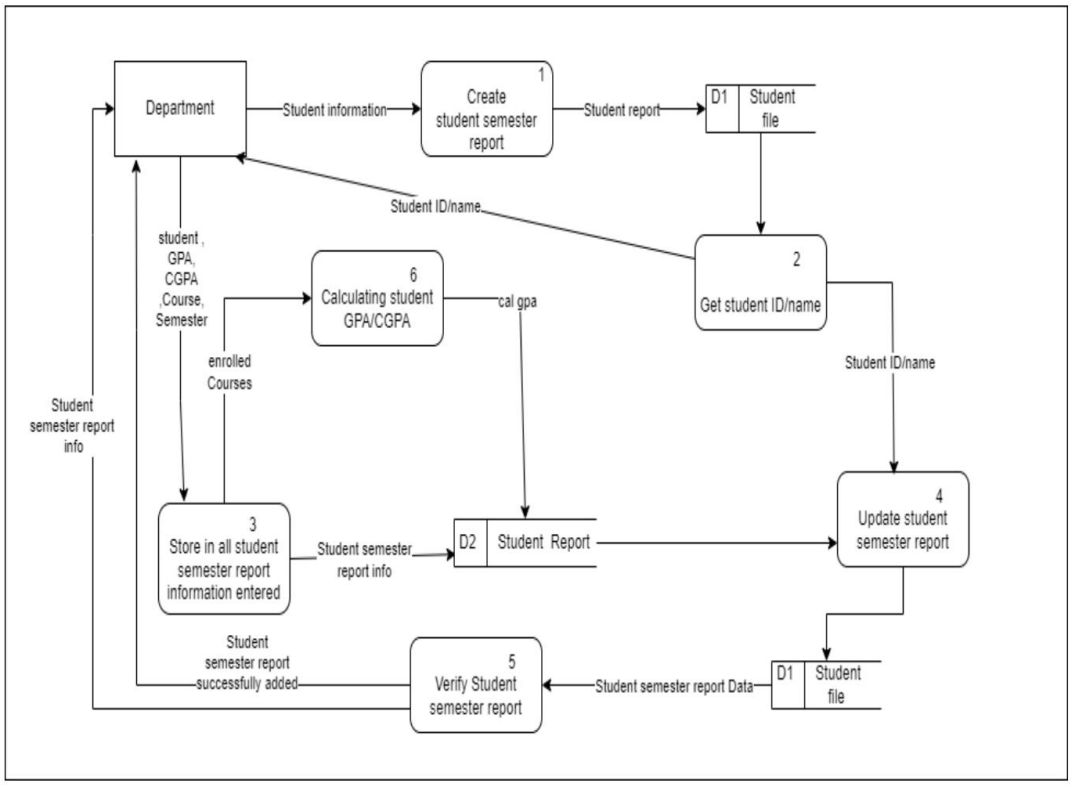

sd:Edit performance
Doctor
Controller management System
Database management system
EnterLogIn(Username,password)
VerfiyUserInformationFromDatabase(Username,Password)
Login result
ALT
Login Successfully
Login failed
RequestEditPerformance()
VerfiyEditPerformanceRequest()
ReturnRequest()
DisplayEditPerformanceOptions()
ALT
RequestEditCourseMark(attendance,course,marks,stu)
VerifyEditCourseMark(attendance,course,marks,stu)
storeineditedsemestermarks (stu, course,marks)
Confirmation result
ALT
successfully added
unsuccessful, wrong info entered
RequestEditCoursesAttendance(attendance,course,stu)
VerifyEditCourseAttendance(attendance,course,marks,stu)
storeineditedsemesterattendance (attendance,course,marks,stu)
Confirmation result
ALT
successfully added
unsuccessful, wrong info entered


sd:Create salary for doctor or TA
Department
Controller Management
Database Management system
EnterLogIn(Username,password)
VerifyUserInformationFromDatabase(Username,Password)
LogIn result
ALT
LogIn Successfully
LogIn failed
ALT
RequestCreateSalaryReportForDR(Salary ,doc,month,Vat)
VerifySalaryReportEnteredInfo(Salary ,doc,month,Vat)
SalaryReportInformationResult()
ALT
InformationNotCorrect()
RequestStoreReportGivenInfo(Salary ,doc,month,Vat)
StoreIn(Salary ,doc,month,Vat)
StoreInVerify()
SuccessfullyAddedSalaryReportForDr()
RequestCreateSalaryReportForTA(Salary ,TA,month,Vat)
VerifySalaryReportEnteredInfo(Salary ,TA,month,Vat)
SalaryReportInformationResult()
ALT
InformationNotCorrect()
RequestStoreReportGivenInfo(Salary ,TA,month,Vat)
StoreIn(Salary ,TA,month,Vat)
StoreInVerify()
SuccessfullyAddedSalaryReportForTA()

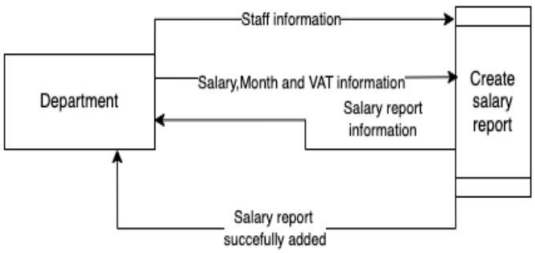

sd :payment
Student
controller Managment
Database Managment System
Bank Authorize
Login(UserName,Password)
VerifyUserInformationFromDataBase(UserName,Password)
LoginResult
alt
Login Successfully
login Failed
RequestForPayingForSemesterFees
request payment for semester details()
request payment for semester details()
alt
insert credit card info
store credit card info
store credit card info
Authorize card Info()
Authorize Credit()
authorization result()
alt
Successfull card info
Failed process
store account info
Insert Bank account
Store account info
Authorize Bank Account
Authorize bankaccount info()
Authorize Result()
alt
Successfull Bankaccount Info
Failed Process

7. The data flow diagrams:

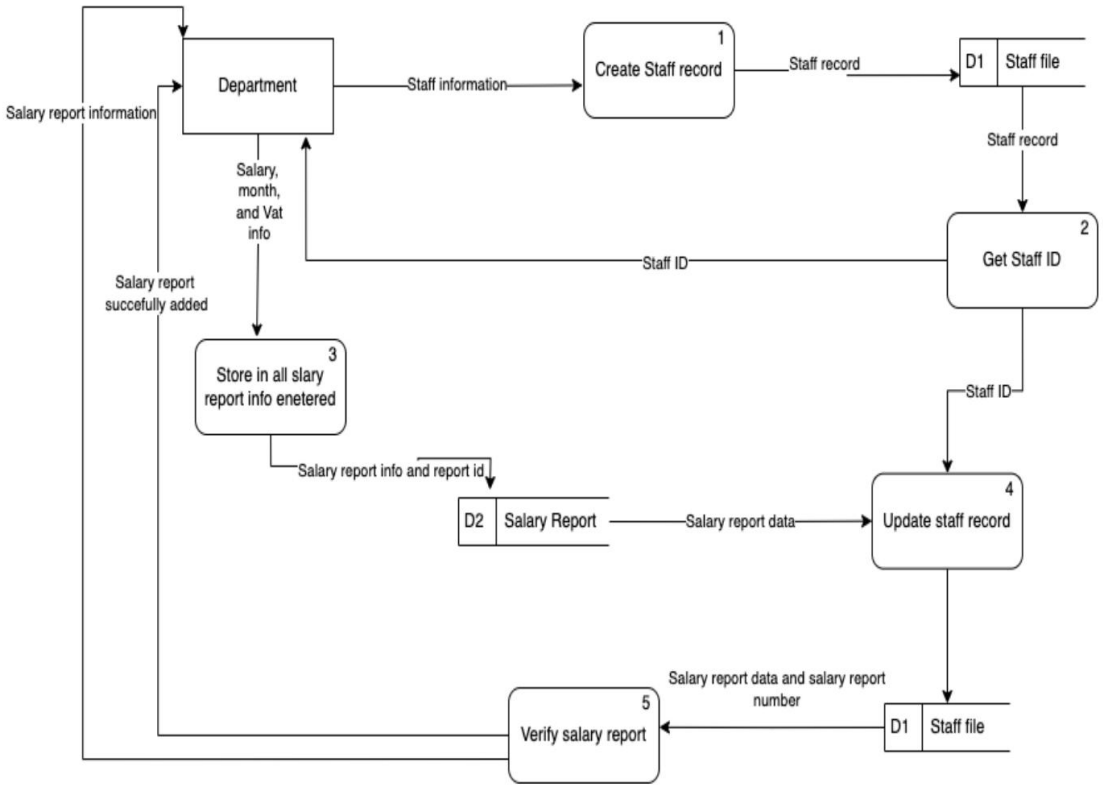


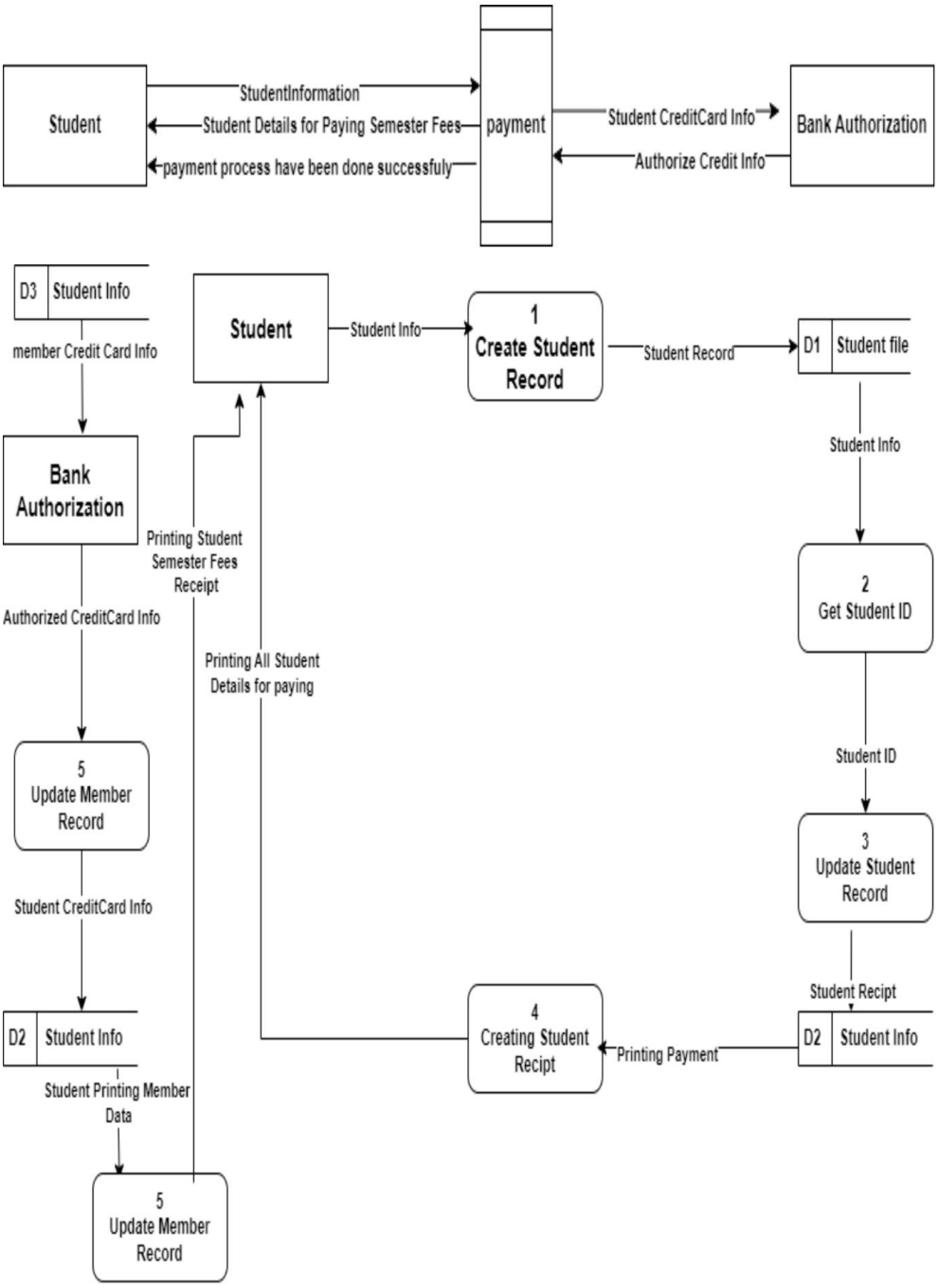


Level 0 Data flow for creating salary report for a staff member

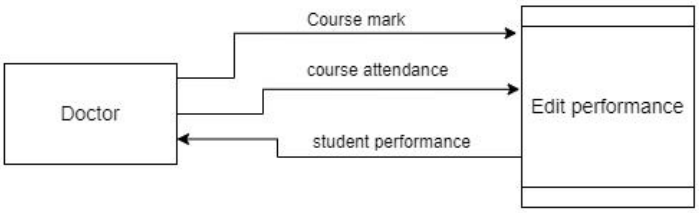


Level 1 Data flow for creating salary report for a staff member

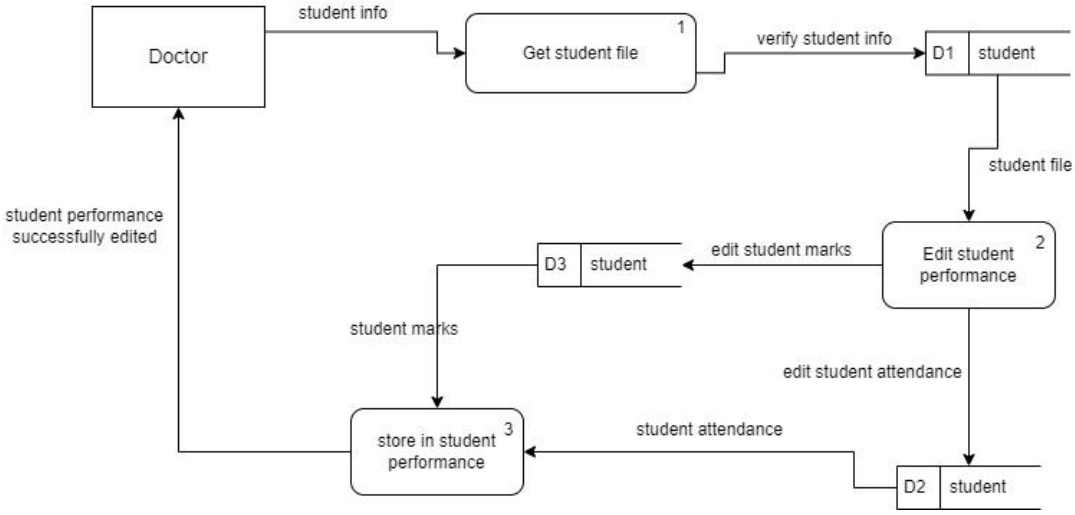


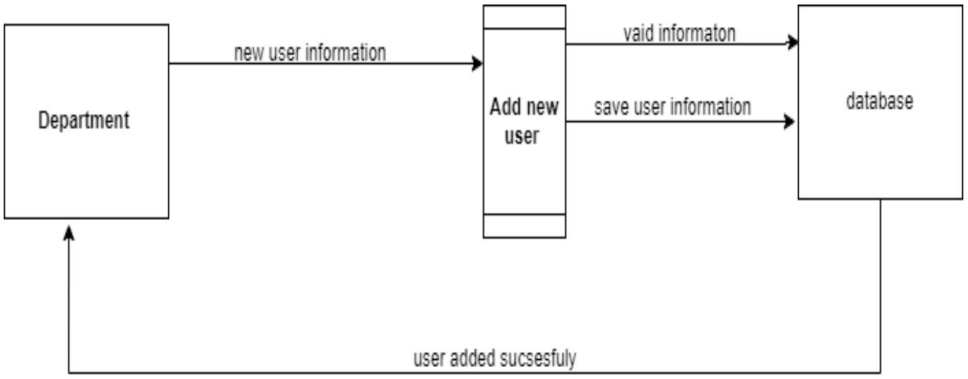

Student
StudentInformation
Student Details for Paying Semester Fees
payment process have been done successfuly
payment
Student CreditCard Info
Authorize Credit Info
Bank Authorization
D3 Student Info
member Credit Card Info
Bank Authorization
Authorized CreditCard Info
5 Update Member Record
Student CreditCard Info
D2 Student Info
Student Printing Member Data
5 Update Member Record
Printing Student Semester Fees Receipt
Printing All Student Details for paying
Student
Student Info
1 Create Student Record
Student Record
D1 Student file
Student Info
2 Get Student ID
Student ID
3 Update Student Record
Student Recipt
D2 Student Info
Printing Payment
4 Creating Student Recipt

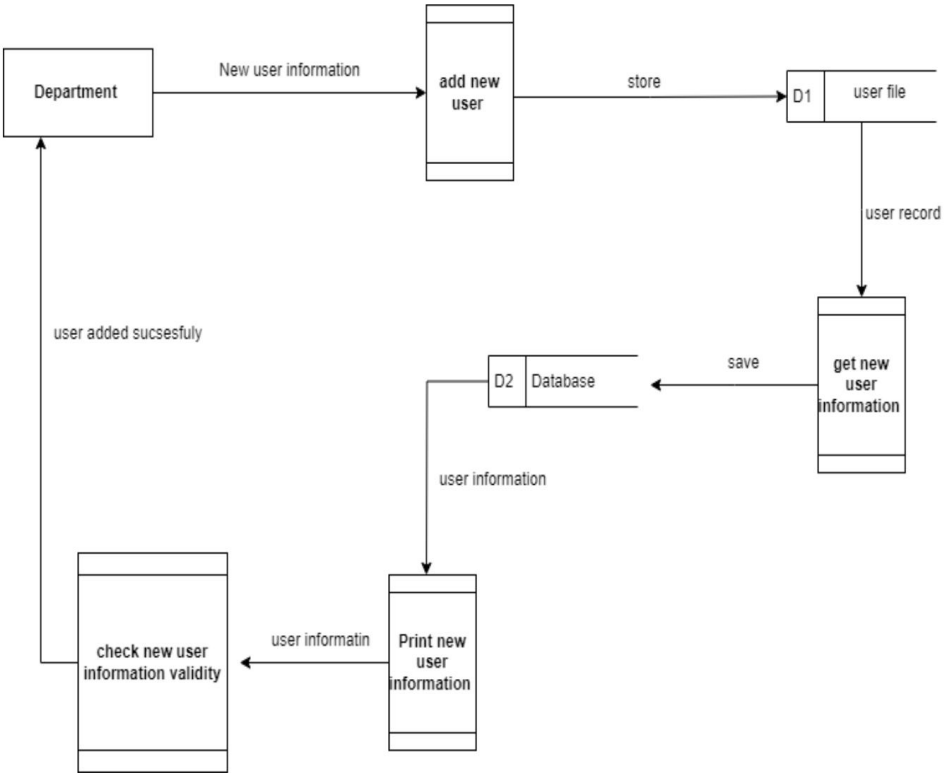
Level 0 : Edit performance



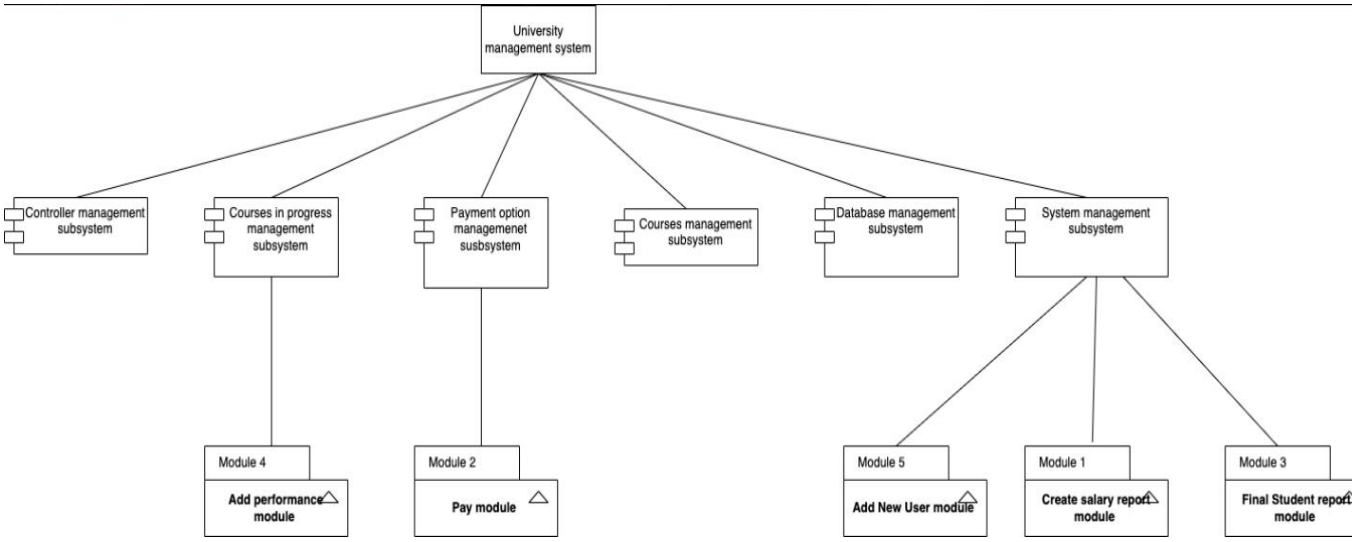
Level 1 : Edit performance




Department
new user information
Add new user
vaid informaton
save user information
database
user added sucsesfuly


Department
New user information
add new user
store
D1
user file
user record
user added sucsesfuly
save
D2
Database
get new user information
user information
check new user information validity
user informatin
Print new user information

Our System:



Planing:

For our system testing, we are planning to use a top to bottom approach as it would be a better approach for lots of prototyping, and since our system is mainly about taking data and processing it into some meaningful information, it would be easy to isolate each system if the given data into the system is meaningful.

- **Unit Test Plan:**

Unit test 1-->module 1
Unit test 2-->module 2
Unit test 3-->module 3
Unit test 4-->module 4
Unit test 5-->module 5

- **Regression Test Plan:**

Regression test 1-->module 1
Regression test 2-->module 2
Regression test 3-->module 3
Regression test 4-->module 4
Regression test 5-->module 5

- **Integration Test Plan:**

Integration Test 1--> module 1 & 2
Integration Test 2-->module 3 & 4
Integration Test 3-->module 1 & 2 & 3 & 4 & 5

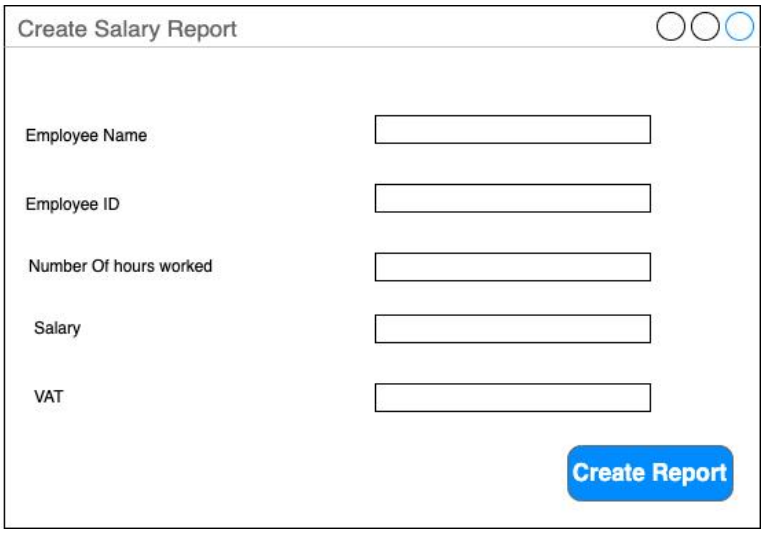
- **Overall Test Plan:**

Unit test 1-->module 1
Unit test 2-->module 2
Regression test 1-->module 1
Regression test 2-->module 2

Integration Test 1--> module 1 & 2
Unit test 3-->module 3
Unit test 4-->module 4
Regression test 3-->module 3
Regression test 4-->module 4
Integration Test 2-->module 3 & 4
Unit test 5-->module 5
Regression test 5-->module 5
Integration Test 3-->module 1 & 2 & 3 & 4 & 5

3)Unit test:

Unit Test 1 for Create salary reports Module:



Create Salary Report module unit test for employee name:

Test Requirements	Test case no	Values	Description
With valid entered name	1	"Ayten Hossam Ahmed"	accept
With invalid entered name	2	"Ayten &Hossam"	Error message--> Accept with no special characters
With invalid entered name	3	"Ayten12"	Error message--> Accept with no special numbers
With invalid entered name	4	"Ayten0.5"	Error message--> Accept with no decimal numbers
With invalid entered name	5	"ayten hossam"	Error message--> Accept with capital letters and small letters

Create Salary Report module unit test for employee id:

Test Requirements	Test case no	Values	Description
With valid entered name	7	21100785	accept
With invalid entered name	8	21100ay	Error message--> Accept with no characters only numbers
With invalid entered name	9	2111888899999	Error message--> Accept with only 8 numbers
With invalid entered name	10	21199@	Error message--> Accept with no Special characters
With invalid entered name	11	-21333	Error message--> Accept only postive numbers
With invalid entered name	12	343444	Error message--> Employee id nor stored in database

Create Salary Report module unit test for employee number of worked days:

Test Requirements	Test case no	Values	Description
With valid entered name	13	30	accept
With invalid entered name	14	500	Error message--> Accept with only 2 numbers
With invalid entered name	15	-2	Error message--> Accept with only positive numbers
With invalid entered name	16	5t	Error message--> Accept with no characters only numbers
With invalid entered name	17	56*	Error message--> Accept with no special characters only numbers
With invalid entered name	18	0.8	Error message--> Accept with no decimal numbers

Create Salary Report module unit test for employee number of hours:

Test Requirements	Test case no	Values	Description
With valid	19	800	accept

With invalid	20	5008	Error message--> Accept with only 3 numbers
With invalid	21	-67	Error message--> Accept with only positive numbers
With invalid	22	88ii	Error message--> Accept with no characters only numbers
With invalid	23	56&	Error message--> Accept with no special characters only numbers
With invalid	24	66.9	Error message--> Accept with no decimal numbers

Create Salary Report module unit test for employee salary:

Test Requirements	Test case no	Values	Description
With valid	25	50000	accept
With invalid	26	790800000	Error message--> Accept with only max 6 numbers
With invalid	27	-878787	Error message--> Accept with only positive numbers
With invalid	28	90000jij	Error message--> Accept with no characters only numbers
With invalid	29	9076544%	Error message--> Accept with no special characters only numbers
With invalid	30	100008	Error message--> Accept with only number<100000 And <1000

Create Salary Report module unit test for employee VAT:

Test Requirements	Test case no	Values	Description
With valid	31	1680 where salary 12000	accept
With invalid	32	5008 where salary 400	Error message--> Accept with the 14% value of salary
With invalid	33	-67909 where salary 10000	Error message--> Accept with only positive numbers

With invalid	34	78787bb where salary 4000	Error message--> Accept with no characters only numbers
With invalid	35	56& where salary 7000	Error message--> Accept with no special characters only numbers
With invalid	30	100008 where salary 8000	Error message--> Accept with only number<100000 And <1000

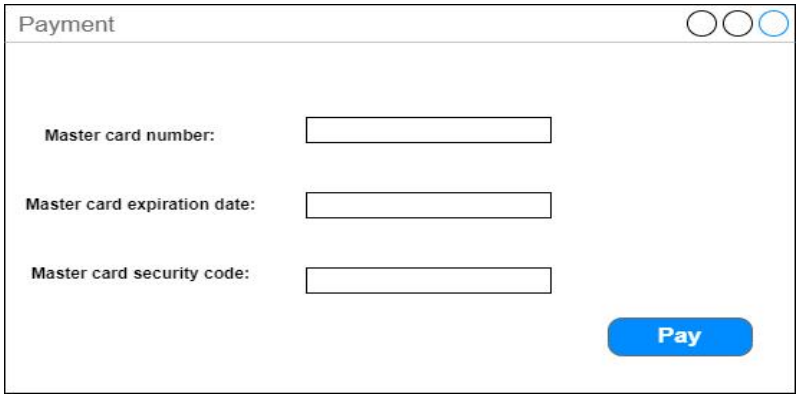
Create Salary Report module unit test for creat report button:

Test Requirements	Test case no	action	Description
With valid	31	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 21100898, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	accept
With invalid	32	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 21100808, Nmuber of days 1000,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID doesn't exists in database,Number of days more than 3 characters
With invalid	33	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 211, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID less than 8 characters

Create Salary Report module unit test for cancel button:

Test Requirements	Test case no	action	Description
Valid data or invalid data	34	Click cancel button	Accept(clear all entered data)

Unit Test 2 : payment:



Payment module unit test for Master Card Number

Test Requirements	Test case no	Values	Description
With Valid entered master card number	1	6789345729786	Accept
With Invalid entered master card number	2	5.3 2.3 5.4 5.2 3.3 2.1 4.1 3.5 7.5 3.6 6.6 3.9	Error message--> Accept with no decimal numbers
With Invalid entered master card number	3	526373	Error message--> Accept when it contains 12 characters
With Invalid entered master card number	4	-678934572	Error message--> Accept only positive numbers
With Invalid entered master card number	5	(6,5,6,8,9,6)	Error message--> Accept with no Special characters
With Invalid entered master card number	6	3647Aysha	Error message--> Accept with no Letters
With Invalid entered master card number	7	2 4 5 7 8 4 8 4	Error message--> Accept with no Spaces

Payment module unit test for Master Card Expiration Date

Test Requirements	Test case no	Values	Description
With Valid entered master card expiration date	8	22/10	Accept
With invalid entered master card expiration date	9	22/2/3/5	Error message--> Accept when it contains only 5 characters
With invalid entered master card expiration date	10	2.22/0.5	Error message--> Accept with no decimal numbers
With invalid entered master card expiration date	11	-2/-10	Error message--> Accept only positive numbers
With invalid entered master card expiration date	12	2@$3	Error message--> Accept with no Special characters
With invalid entered master card expiration date	13	3Abc5	Error message--> Accept with no Letters
With invalid entered master card expiration date	14	3 4 6 8	Error message--> Accept with no Spaces

Payment module unit test for Master Card Security Code

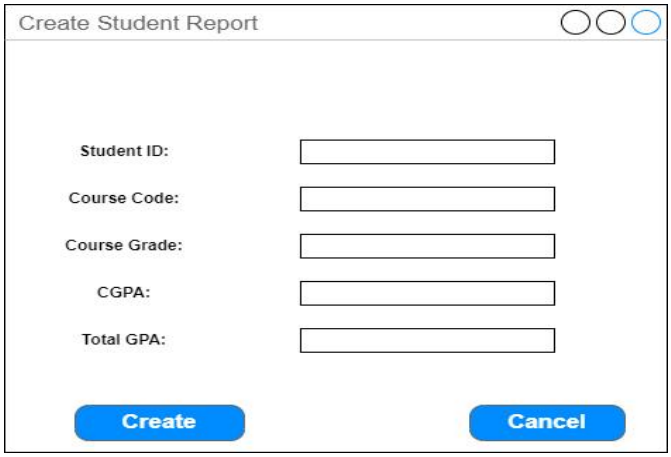
Test Requirements	Test case no	Values	Description
With Valid entered master card security code	15	1234567	Accept

With invalid entered master card security code	16	123456789	Error message--> Accept when it contains only 7 characters
With invalid entered master card security code	17	123456 6.6	Error message--> Accept with no decimal numbers
With invalid entered master card security code	18	-1234567	Error message--> Accept only positive numbers
With invalid entered master card security code	19	2@$34567	Error message--> Accept with no Special characters
With invalid entered master card security code	20	3Abc5567	Error message--> Accept with no Letters
With invalid entered master card security code	21	3 4 6 8 1 2 4 5	Error message--> Accept with no Spaces

Payment module unit test for Pay button

Test Requirements	Test case no	Values	Description
valid data	22	Click pay button after entering master card number 6789345729786, master card expiration date 22/10, master card security code 1234567	Accept
Invalid data	23	Click pay button after entering master card number 6789@34572$9786, master card expiration date 2$@1, master card security code 1234567	Reject Error message--> Master card number contains special characters,Master card expiration date contains special characters
Invalid data	24	Click pay button after entering master card number 678AB457f9786, master card expiration date 6 8 2 / 1 0, master card security code 1234567	Reject Error message--> Master card number contains Letters,Master card expiration date contains spaces

Unit Test 3: Create student's semester report:



Create Student Report Module unit test:

For Course Code:

Test Requirements	Test Case no	Values	Description
With valid code	1	CSE251	Accept
With invalid code	2	CSE251$	Error Message ---> Accept only letters and numbers
With invalid code	3	CSE 251	Error Message --> No Space

For Course Grade:

Test Requirements	Test Case no	Values	Description
With valid Grade	4	A/B/C	Accept
With valid Grade	5	C-/C	Accept
With invalid Grade	6	A123	Error Message --> No Numbers
With invalid Grade	7	12.3	Error Message--> Accept letters only
With invalid Grade	8	A$/B#	Error Message--> Accept letter only

For GPA/CGPA:

Test Requirements	Test Case no	Values	Description
With valid GPA/CGPA	9	3.2/3	Accept
With invalid GPA/CGPA	10	-3.3/-3	Error Message --> Only positive numbers
With invalid GPA/CGPA	11	3A/A	Error Message--> Accept Numbers only
With invalid GPA/CGPA	12	5/6	Error Message--> Accept < 4 only
With invalid GPA/CGPA	13	<3>/3.	Error Message--> No character only numbers

For Student ID:

Test Requirements	Test case no	Values	Description
With valid ID	14	21100824	accept
With invalid ID	15	21100ay	Error message--> Accept with no characters only numbers
With invalid ID	16	2111888899999	Error message--> Accept with only 8 numbers
With invalid ID	17	21199@	Error message--> Accept with no Special characters

With invalid ID	18	-21333	Error message--> Accept only positive numbers
With invalid ID	19	343444	Error message--> Student id not stored in database

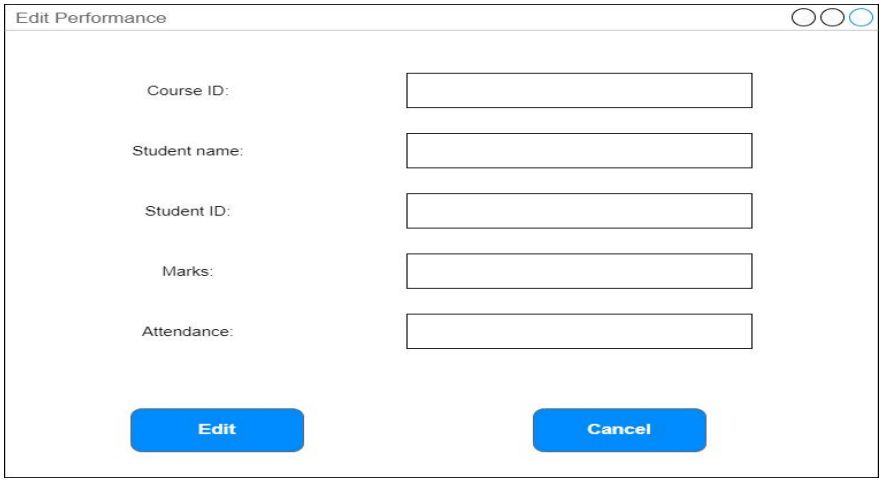
For Create Component:

Test Requirements	Test case no	Action	Description
With valid data	20	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGPA/GPA less than or equal 4	Accept
With invalid data	21	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGOA/GPA more than 4	Reject

For the Cancel Component:

Test Requirements	Test case no	Action	Description
Valid data or invalid data	22	Click Cancel button	Accept (clear all entered data)

Unit Test 4: Add/Edit Performance:



Course ID unit test :

Test Requirement	Test Case Number	Values	Description
Valid value	1	B130	Accept and already stored in database
Invalid value	2	:89;	Error message: Contain characters
Invalid value	3	-O300	Error message: Can't be negative number
Invalid value	4	t087	Error message: Small letter
Invalid value	5	"blank"	Error message: Enter value

Student Name unit test:

Test Requirement	Test Case Number	Values	Description
Valid value	6	“Mariam Ahmed tamer”	Accept
Invalid value	7	Hana342	Error message: Contain number
Invalid value	8	Hana=-	Error message: Contain character
Invalid value	9	MariamAhmed	Error message: No spaces
Invalid value	10	“blank”	Error message: Enter value

Student ID unit test:

Test Requirement	Test Case Number	Values	Description
Valid value	11	22100534	Accept
Invalid value	12	211[#86=	Error message: Contain character
Invalid value	13	211AbcD67	Error message: Contain letters
Invalid value	14	-22100534	Error message: Negative value
Invalid value	15	“blank”	Error message: Enter value
Invalid value	16	2110086387	Error message: Exceeded 8 numbers

Marks unit test:

Test Requirement	Test Case Number	Values	Description
Valid value	17	83	Accept
Invalid value	18	)(86+	Error message: Contain character
Invalid value	19	cvER67	Error message: Contain letters
Invalid value	20	-98	Error message: Negative value
Invalid value	21	“blank”	Error message: Enter value
Invalid value	22	9573	Error message: Exceeded 3 numbers

Attendance unit test:

Test Requirement	Test Case Number	Values	Description
Valid value	23	6	Accept
Invalid value	24	[#=9	Error message: Contain character
Invalid value	25	AbcD67	Error message: Contain letters
Invalid value	26	-8	Error message: Negative value
Invalid value	27	“blank”	Error message: Enter value
Invalid value	28	4532	Error message: Exceeded 2 numbers

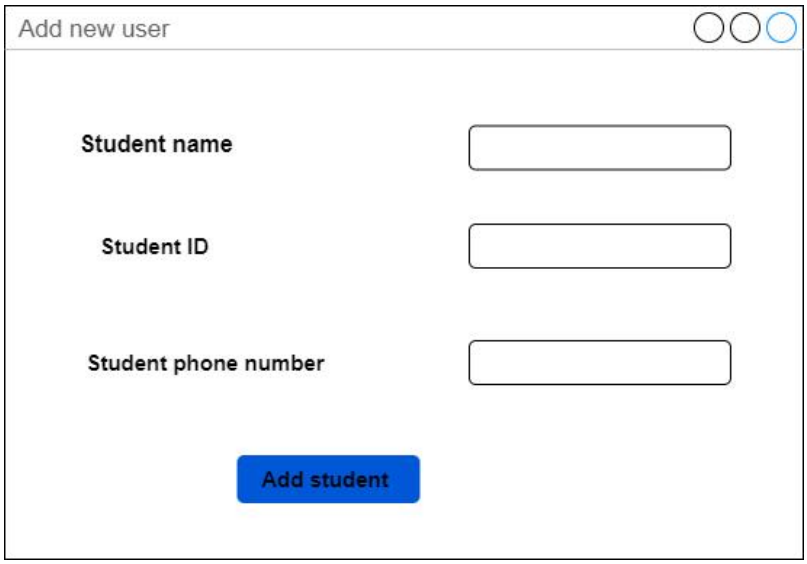
Edit button unit test:

Test Requirement	Test Case Number	Values	Description
Valid value	29	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Accept
Invalid value	30	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100#*+ , Marks 99,Attendance 900	Error message: Invalid Student ID and Attendance
Invalid value	31	Click edit button after entering Course ID D8092,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Error message: Invalid course ID

Cancel button unit test:

Test Requirement	Test Case Number	Values	Description
Valid or invalid	32	Click cancel button	Accept and clears all data

Unit Test 5: Add new user:



Student Name component unit test :

Test Requirement	Test Case Number	Values	Description

Valid	1	Ahmed Elsayed	Accepted and saved
invalid	2	Ahmed ELSAYED @	Error massage : cant contain any special characters “&*#$@’\?|<>,.”
invalid	3	Ahmed Elsayed 11	Error massage : cant contain any numbers”123456789”
invalid	4	AhmedELsayed	Error massage:There is no space between first name and last name

Student ID component unit test:

Test Requirement	Test Case Number	Values	Description
With valid entered name	5	21100818	accept
With invalid entered name	6	21100AH	Error message--> Accept with no characters onl numbers
With invalid entered name	7	21118188888888888889	Error message--> Accept with only 8 numbers
With invalid entered name	8	21199#	Error message--> Accept with no Special characters

Student Phone Number component unit test:

With valid entered name	9	01017230586	accept
With invalid entered name	10	01017230586A	Error message--> Accept with only numbers
With invalid entered name	11	010172305	Error message--> Accept 12 numbers
With invalid entered name	12	010172305866666	Error message--> More than 12 numbers

With invalid entered name	13	01017230586&	Error message--> Accept with no special charact only numbers
With invalid entered name	14	01017230586.4	Error message--> Accept with no decimal numb

Add Student button component unit test:

Test Requirement	Test Case Number	Values	Description
valid	15	**Click save** Name:Ahmed Elsayed ID:21100818 Phone :01017230586	Accept and saved to databas
invalid	16	**Click save** Name:Ahmed ELSAYED @ ID:21100AH Phone:01017230586A	Error message--> Accept with no characters only numbers
invalid	17	**Click save** Name:Ahmed Elsayed 11 ID:21118188888888888889 Phone:010172305	Error message--> Accept with only 8 numbers
invalid	18	**Click save** Name:AhmedELsayed ID:21199# Phone :01017230586.4	Error message--> Accept with no Special characters

4)Regression testing:

Regression test 1 Create Salary Report module:

Test Requirements	Test case no	Values	Description
With valid entered name	1	"Mai Ahmed"	accept
With invalid entered name	2	"Mai &shawer"	Error message--> Accept with no special characters
With invalid entered name	3	"Mai992"	Error message--> Accept with no special numbers
With invalid entered name	4	"Mai0.55565"	Error message--> Accept with no decimal numbers
With invalid entered name	5	"mai ahmed"	Error message-->

			Accept with capital letters and small letters
With valid entered name	7	21100798	accept
With invalid entered name	8	888282j	Error message--> Accept with no characters only numbers
With invalid entered name	9	9983837373	Error message--> Accept with only 8 numbers
With invalid entered name	10	23883$	Error message--> Accept with no Special characters
With invalid entered name	11	-8838383	Error message--> Accept only postive numbers
With invalid entered name	12	21100999	Error message--> Employee id nor stored in database
With valid entered name	13	39	accept
With invalid entered name	14	580	Error message--> Accept with only 2 numbers
With invalid entered name	15	-89	Error message--> Accept with only positive numbers
With invalid entered name	16	909u	Error message--> Accept with no characters only numbers
With invalid entered name	17	8484^	Error message--> Accept with no special characters only numbers
With invalid entered name	18	39.8	Error message--> Accept with no decimal numbers
With valid entered name	19	999	accept
With invalid entered name	20	80909	Error message--> Accept with only 3 numbers
With invalid entered name	21	-89898	Error message--> Accept with only positive numbers
With invalid entered name	22	777as	Error message--> Accept with no characters only numbers
With invalid entered name	23	7474;;	Error message--> Accept with no special characters only numbers
With invalid entered name	24	150.8	Error message--> Accept with no decimal numbers
With valid entered name	25	59000	accept

With invalid entered name	26	3983373730	Error message--> Accept with only max 6 numbers
With invalid entered name	27	-90909	Error message--> Accept with only positive numbers
With invalid entered name	28	74746466nn	Error message--> Accept with no characters only numbers
With invalid entered name	29	9737373%	Error message--> Accept with no special characters only numbers
With invalid entered name	30	1828288	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	1960 where salary 14000	accept
With invalid entered name	32	13448 where salary 5000	Error message--> Accept with the 14% value of salary
With invalid entered name	33	-45569 where salary 99000	Error message--> Accept with only positive numbers
With invalid entered name	34	1234lb where salary 9090	Error message--> Accept with no characters only numbers
With invalid entered name	35	500^ where salary 9000	Error message--> Accept with no special characters only numbers
With invalid entered name	30	7000000 where salary 90000	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 21100898, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	accept
With invalid entered name	32	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 21100808, Nmuber of days 1000,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID doesn’t exists in database,Number of days more than 3 characters
With invalid entered name	33	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 211,	Reject Error message--> Employee ID less than 8 characters

		Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	
Valid data or invalid data	34	Click cancel button	Accept(clear all entered data)

Regression test2 Payment module :

Test Requirements	Test case no	Values	Description
With Valid entered master card number	1	123456789123	Accept
With Invalid entered master card number	2	4.44323567829	Error message--> Accept with no decimal numbers
With Invalid entered master card number	3	678539	Error message--> Accept when it contains 12 characters
With Invalid entered master card number	4	-983135974	Error message--> Accept only positive numbers
With Invalid entered master card number	5	%123@^58	Error message--> Accept with no Special characters
With Invalid entered master card number	6	Mariam2341	Error message--> Accept with no Letters
With Invalid entered master card number	7	5 6 7 8 9 3 2	Error message--> Accept with no Spaces
With Valid entered master card expiration date	8	10/10	Accept
With invalid entered master card expiration date	9	12/2/6/8	Error message--> Accept when it contains only 5 characters
With invalid entered master card expiration date	10	0.6/7.2	Error message--> Accept with no decimal numbers
With invalid entered master card expiration date	11	-6/-11	Error message--> Accept only positive numbers
With invalid entered master card expiration date	12	2%&12	Error message--> Accept with no Specialcharacters
With invalid entered master card expiration date	13	5SfH7	Error message--> Accept with no Letters
With invalid enteredmaster card expiration date	14	5 7 8 9 0	Error message--> Accept with no Spaces
With Valid entered master card security code	15	5678392	Accept
With invalid entered master card security code	16	5678945632	Error message--> Accept when it contains only 7 characters

With invalid entered master card security code	17	6.6453327	Error message--> Accept with no decimal numbers
With invalid entered master card security code	18	-6799752	Error message--> Accept only positive numbers
With invalid entered master card security code	19	2@%$789	Error message--> Accept with no Special characters
With invalid entered master card security code	20	4D7G8hk8	Error message--> Accept with no Letters
With invalid entered master card security code	21	6 7 8 0 5 4 8	Error message--> Accept with no Spaces
valid data	22	Click pay button after entering master card number 123456789123, master card expiration date 10/10, master card security code 5678392	Accept
Invalid data	23	Click pay button after entering master card number 1233@34572$9786, master card expiration date 2$@1, master card security code 1267867	Reject Error message--> Master card number contains special characters,Master card expiration date contains special characters
Invalid data	24	Click pay button after entering master card number 78AB457f9786, master card expiration date 6 8 7 / 1 3, master card security code 5678953	Reject Error message--> Master card number contains Letters,Master card expiration date contains spaces

Regression Test3 Create Student Report Module:

Test Req	Test Case no	Component	Values	Description
With valid code	1	***Course code***	CSE221	Accept
With invalid code	2	***Course code***	CSE221$	Error Message ---> Accept only letters and numbers
With invalid code	3	***Course Code***	CSE 221	Error Message --> No Space
With valid Grade	4	***Course Grade***	D/F	Accept
With valid Grade	5	***Course Grade***	D-/D	Accept
With invalid Grade	6	***Course Grade***	B13	Error Message --> No Numbers
With invalid Grade	7	***Course Grade***	15.344	Error Message--> Accept letters only

With invalid Grade	8	***Course Grade***	F@/C&	Error Message--> Accept letter only
With valid GPA/CGPA	9	***GPA/CGPA***	1.3/1	Accept
With invalid GPA/CGPA	10	***GPA/CGPA***	-2.3/-2	Error Message --> Only positive numbers
With invalid GPA/CGPA	11	***GPA/CGPA***	2B/B	Error Message--> Accept Numbers only
With invalid GPA/CGPA	12	***GPA/CGPA***	4.5/5	Error Message--> Accept < 4 only
With invalid GPA/CGPA	13	***GPA/CGPA***	<2>/1.	Error Message--> No character only numbers
With valid ID	14	***Student ID***	21100822	accept
With invalid ID	15	***Student ID***	20100bg	Error message--> Accept with no characters only numbers
With invalid ID	16	***Student ID***	2111000099469	Error message--> Accept with only 8 numbers
With invalid ID	17	***Student ID***	20087#	Error message--> Accept with no Special characters
With invalid ID	18	***Student ID***	-21100824	Error message--> Accept only positive numbers
With invalid ID	19	***Student ID***	3992456	Error message--> Student id not stored in database
			Action	**Description**
With valid data	20	***Create***	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGPA/GPA less than or equal 4	Accept
With invalid data	21	***Create***	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGOA/GPA more than 4	Reject
Valid data or invalid data	22	Cancel	Click Cancel button	Accept (clear all entered data)

Regression test 4 Edit performance module:

Test Requirement	Test Case Number	Values	Description
Valid value	1	d140	Accept and already stored in database
Invalid value	2	:09;	Error message: Contain characters
Invalid value	3	C-300	Error message: Can't be negative number
Invalid value	4	t087	Error message: Small letter
Invalid value	5	"blank"	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	6	"Yousef Ahmed Kamel"	Accept

Invalid value	7	Ahmed342	Error message: Contain number
Invalid value	8	Ahmed=-	Error message: Contain character
Invalid value	9	YousefAhmed	Error message: No spaces
Invalid value	10	“blank”	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	11	22100818	Accept
Invalid value	12	211[#86=	Error message: Contain character
Invalid value	13	211AbcD00	Error message: Contain letters
Invalid value	14	-22100818	Error message: Negative value
Invalid value	15	“blank”	Error message: Enter value
Invalid value	16	21100863818	Error message: Exceeded 8 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	17	99	Accept
Invalid value	18	)(96+	Error message: Contain character
Invalid value	19	cvER67	Error message: Contain letters
Invalid value	20	-90	Error message: Negative value
Invalid value	21	“blank”	Error message: Enter value
Invalid value	22	1862	Error message: Exceeded 3 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	23	1	Accept
Invalid value	24	[#=9	Error message: Contain character
Invalid value	25	AbcD67	Error message: Contain letters
Invalid value	26	-7	Error message: Negative value
Invalid value	27	“blank”	Error message: Enter value
Invalid value	28	9265	Error message: Exceeded 2 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	29	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Accept
Invalid value	30	Click edit button after entering Course ID D100,Student Name	Error message: Invalid Student ID and Attendance

		Yousef Ahmed Kamel ,Student ID 21100#*+ , Marks 99,Attendance 900	
Invalid value	31	Click edit button after entering Course ID D8092,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Error message: Invalid course ID
Test Requirement	Test Case Number	Values	Description
Valid or invalid	32	Click cancel button	Accept and clears all data

Regression Test 5 Add New User Module:

Test Requirement	*Test Case Number*	Component	*Values*	Description
Valid	1	User name	Hana Mohamed	Accepted and saved
invalid	2	User name	Hana Mohamed @	Error massage : cant contain any special characters "&*#$@'\?|<>,."
invalid	3	User name	Hana Mohamed 11	Error massage : cant contain any numbers"123456789"
invalid	4	User name	HanaMohamed	Error massage:There is no space between first name and last name
Test Requirement	*Test Case Number*	*Values*	Description	*Test Requirement*
With valid entered name	5	21100863	accept	With valid entered name
With invalid entered name	6	21100GHH	Error message--> Accept with no characters only numbers	With invalid entered name
With invalid entered name	7	21118777777777777779	Error message--> Accept with only 8	With invalid entered name

			numbers	
With invalid entered name	8	21177#	Error message--> Accept with no Special characters	With invalid entered name
With valid entered name	9	01003471736	accept	With valid entered name
With invalid entered name	10	01003471736ooA	Error message--> Accept with only numbers	With invalid entered name
With invalid entered name	11	010034717	Error message--> Accept 12 numbers	With invalid entered name
With invalid entered name	12	010034717369999999	Error message--> More than 12 numbers	With invalid entered name
With invalid entered name	13	01003471736@#	Error message--> Accept with no special characters only numbers	With invalid entered name
With invalid entered name	14	01003471736.55	Error message--> Accept with no decimal numbers	With invalid entered name
Test Requirement	*Test Case Number*	*Values*	Description	*Test Requirement*
valid	15	**Click save** Name: Hana Mohamed ID:21100863 Phone : 01003471736	Accept and saved to database	valid
invalid	16	**Click save** Name: : Hana Mohamed #$ ID:21100SD Phone:01017230LLL	Error message--> Accept with no characters only numbers	invalid
invalid	17	**Click save** Name: Hana Mohamed 11 ID: 2110086333333	Error message-->	invalid

		Phone:01014405	Accept with only 8 numbers	
invalid	18	**Click save** Name: HanaMohamed ID:211008@@ Phone : 01003471736.55	Error message--> Accept with no Special characters	invalid

5) Integration Test:

Integration test 1 for module 1 & 2:

Test Requirements	Test case no	Values	Description
With valid entered name	1	"Mai Ahmed"	accept
With invalid entered name	2	"Mai &shawer"	Error message--> Accept with no special characters
With invalid entered name	3	"Mai992"	Error message--> Accept with no special numbers
With invalid entered name	4	"Mai0.55565"	Error message--> Accept with no decimal numbers
With invalid entered name	5	"mai ahmed"	Error message--> Accept with capital letters and small letters
With valid entered name	7	21100798	accept
With invalid entered name	8	888282j	Error message--> Accept with no characters only numbers
With invalid entered name	9	9983837373	Error message--> Accept with only 8 numbers
With invalid entered name	10	23883$	Error message--> Accept with no Special characters
With invalid entered name	11	-8838383	Error message--> Accept only postive numbers
With invalid entered name	12	21100999	Error message--> Employee id nor stored in database
With valid entered name	13	39	accept

With invalid entered name	14	580	Error message--> Accept with only 2 numbers
With invalid entered name	15	-89	Error message--> Accept with only positive numbers
With invalid entered name	16	909u	Error message--> Accept with no characters only numbers
With invalid entered name	17	8484^	Error message--> Accept with no special characters only numbers
With invalid entered name	18	39.8	Error message--> Accept with no decimal numbers
With valid entered name	19	999	accept
With invalid entered name	20	80909	Error message--> Accept with only 3 numbers
With invalid entered name	21	-89898	Error message--> Accept with only positive numbers
With invalid entered name	22	777as	Error message--> Accept with no characters only numbers
With invalid entered name	23	7474;;	Error message--> Accept with no special characters only numbers
With invalid entered name	24	150.8	Error message--> Accept with no decimal numbers
With valid entered name	25	59000	accept
With invalid entered name	26	3983373730	Error message--> Accept with only max 6 numbers
With invalid entered name	27	-90909	Error message--> Accept with only positive numbers
With invalid entered name	28	74746466nn	Error message--> Accept with no characters only numbers
With invalid entered name	29	9737373%	Error message--> Accept with no special characters only numbers
With invalid entered name	30	1828288	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	1960 where salary 14000	accept
With invalid entered name	32	13448 where salary 5000	Error message--> Accept with the 14% value of salary

With invalid entered name	33	-45569 where salary 99000	Error message--> Accept with only positive numbers
With invalid entered name	34	1234lb where salary 9090	Error message--> Accept with no characters only numbers
With invalid entered name	35	500^ where salary 9000	Error message--> Accept with no special characters only numbers
With invalid entered name	30	7000000 where salary 90000	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 21100898, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	accept
With invalid entered name	32	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 21100808, Nmuber of days 1000,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID doesn't exists in database,Number of days more than 3 characters
With invalid entered name	33	Click create report button after entering the Employee name "Bardes Khalid", Employee ID 211, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID less than 8 characters
Valid data or invalid data	34	Click cancel button	Accept(clear all entered data)

Test Requirements	Test case no	Values	Description
With Valid entered master card number	1	123456789123	Accept
With Invalid entered master card number	2	4.44323567829	Error message--> Accept with no decimal numbers
With Invalid entered master card number	3	678539	Error message--> Accept when it contains 12 characters
With Invalid entered master card number	4	-983135974	Error message--> Accept only positive numbers
With Invalid entered master card number	5	%123@^58	Error message--> Accept with no Special characters

With Invalid entered master card number	6	Mariam2341	Error message--> Accept with no Letters
With Invalid entered master card number	7	5 6 7 8 9 3 2	Error message--> Accept with no Spaces
With Valid entered master card expiration date	8	10/10	Accept
With invalid entered master card expiration date	9	12/2/6/8	Error message--> Accept when it contains only 5 characters
With invalid entered master card expiration date	10	0.6/7.2	Error message--> Accept with no decimal numbers
With invalid entered master card expiration date	11	-6/-11	Error message--> Accept only positive numbers
With invalid entered master card expiration date	12	2%&12	Error message--> Accept with no Specialcharacters
With invalid entered master card expiration date	13	5SfH7	Error message--> Accept with no Letters
With invalid enteredmaster card expiration date	14	5 7 8 9 0	Error message--> Accept with no Spaces
With Valid entered master card security code	15	5678392	Accept
With invalid entered master card security code	16	5678945632	Error message--> Accept when it contains only 7 characters
With invalid entered master card security code	17	6.6453327	Error message--> Accept with no decimal numbers
With invalid entered master card security code	18	-6799752	Error message--> Accept only positive numbers
With invalid entered master card security code	19	2@%$789	Error message--> Accept with no Special characters
With invalid entered master card security code	20	4D7G8hk8	Error message--> Accept with no Letters
With invalid entered master card security code	21	6 7 8 0 5 4 8	Error message--> Accept with no Spaces
valid data	22	Click pay button after entering master card number 123456789123, master card expiration date 10/10, master card security code 5678392	Accept
Invalid data	23	Click pay button after entering master card number 1233@34572$9786, master card expiration date 2$@1, master card security code	Reject Error message--> Master card number contains special characters,Master card expiration date contains special characters

		1267867	
Invalid data	24	Click pay button after entering master card number 78AB457f9786, master card expiration date 6 8 7 / 1 3, master card security code 5678953	Reject Error message--> Master card number contains Letters,Master card expiration date contains spaces

Integration test 2 for module 3 & 4:

Test Req	Test Case no	Component	Values	Description
With valid code	1	***Course code***	CSE221	Accept
With invalid code	2	***Course code***	CSE221$	Error Message ---> Accept only letters and numbers
With invalid code	3	***Course Code***	CSE 221	Error Message --> No Space
With valid Grade	4	***Course Grade***	D/F	Accept
With valid Grade	5	***Course Grade***	D-/D	Accept
With invalid Grade	6	***Course Grade***	B13	Error Message --> No Numbers
With invalid Grade	7	***Course Grade***	15.344	Error Message--> Accept letters only
With invalid Grade	8	***Course Grade***	F@/C&	Error Message--> Accept letter only
With valid GPA/CGPA	9	***GPA/CGPA***	1.3/1	Accept
With invalid GPA/CGPA	10	***GPA/CGPA***	-2.3/-2	Error Message --> Only positive numbers
With invalid GPA/CGPA	11	***GPA/CGPA***	2B/B	Error Message--> Accept Numbers only
With invalid GPA/CGPA	12	***GPA/CGPA***	4.5/5	Error Message--> Accept < 4 only
With invalid GPA/CGPA	13	***GPA/CGPA***	<2>/1.	Error Message--> No character only numbers
With valid ID	14	***Student ID***	21100822	accept
With invalid ID	15	***Student ID***	20100bg	Error message--> Accept with no characters only numbers
With invalid ID	16	***Student ID***	2111000099469	Error message--> Accept with only 8 numbers
With invalid ID	17	***Student ID***	20087#	Error message--> Accept with no Special characters
With invalid ID	18	***Student ID***	-21100824	Error message--> Accept only positive numbers
With invalid ID	19	***Student ID***	3992456	Error message--> Student id not stored in database
			Action	**Description**
With valid data	20	***Create***	Click Create button	Accept

			after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGPA/GPA less than or equal 4	
With invalid data	21	***Create***	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGOA/GPA more than 4	Reject
Valid data or invalid data	22	Cancel	Click Cancel button	Accept (clear all entered data)

Test Requirement	Test Case Number	Values	Description
Valid value	1	d140	Accept and already stored in database
Invalid value	2	:09;	Error message: Contain characters
Invalid value	3	C-300	Error message: Can't be negative number
Invalid value	4	t087	Error message: Small letter
Invalid value	5	"blank"	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	6	"Yousef Ahmed Kamel"	Accept
Invalid value	7	Ahmed342	Error message: Contain number
Invalid value	8	Ahmed=-	Error message: Contain character
Invalid value	9	YousefAhmed	Error message: No spaces
Invalid value	10	"blank"	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	11	22100818	Accept
Invalid value	12	211[#86=	Error message: Contain character
Invalid value	13	211AbcD00	Error message: Contain letters
Invalid value	14	-22100818	Error message: Negative value
Invalid value	15	"blank"	Error message: Enter value
Invalid value	16	21100863818	Error message: Exceeded 8 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	17	99	Accept
Invalid value	18	)(96+	Error message: Contain character
Invalid value	19	cvER67	Error message:

			Contain letters
Invalid value	20	-90	Error message: Negative value
Invalid value	21	“blank”	Error message: Enter value
Invalid value	22	1862	Error message: Exceeded 3 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	23	1	Accept
Invalid value	24	[#=9	Error message: Contain character
Invalid value	25	AbcD67	Error message: Contain letters
Invalid value	26	-7	Error message: Negative value
Invalid value	27	“blank”	Error message: Enter value
Invalid value	28	9265	Error message: Exceeded 2 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	29	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Accept
Invalid value	30	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100#*+ , Marks 99,Attendance 900	Error message: Invalid Student ID and Attendance
Invalid value	31	Click edit button after entering Course ID D8092,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Error message: Invalid course ID
Test Requirement	Test Case Number	Values	Description
Valid or invalid	32	Click cancel button	Accept and clears all data

Integration test 3 for module 1,2,3,4& 5:

Test Requirements	Test case no	Values	Description
With valid entered name	1	“Mai Ahmed”	accept
With invalid entered name	2	“Mai &shawer”	Error message--> Accept with no special characters

With invalid entered name	3	“Mai992”	Error message--> Accept with no special numbers
With invalid entered name	4	“Mai0.55565”	Error message--> Accept with no decimal numbers
With invalid entered name	5	“mai ahmed”	Error message--> Accept with capital letters and small letters
With valid entered name	7	21100798	accept
With invalid entered name	8	888282j	Error message--> Accept with no characters only numbers
With invalid entered name	9	9983837373	Error message--> Accept with only 8 numbers
With invalid entered name	10	23883$	Error message--> Accept with no Special characters
With invalid entered name	11	-8838383	Error message--> Accept only postive numbers
With invalid entered name	12	21100999	Error message--> Employee id nor stored in database
With valid entered name	13	39	accept
With invalid entered name	14	580	Error message--> Accept with only 2 numbers
With invalid entered name	15	-89	Error message--> Accept with only positive numbers
With invalid entered name	16	909u	Error message--> Accept with no characters only numbers
With invalid entered name	17	8484^	Error message--> Accept with no special characters only numbers
With invalid entered name	18	39.8	Error message--> Accept with no decimal numbers
With valid entered name	19	999	accept
With invalid entered name	20	80909	Error message--> Accept with only 3 numbers
With invalid entered name	21	-89898	Error message--> Accept with only positive numbers

With invalid entered name	22	777as	Error message--> Accept with no characters only numbers
With invalid entered name	23	7474;;	Error message--> Accept with no special characters only numbers
With invalid entered name	24	150.8	Error message--> Accept with no decimal numbers
With valid entered name	25	59000	accept
With invalid entered name	26	3983373730	Error message--> Accept with only max 6 numbers
With invalid entered name	27	-90909	Error message--> Accept with only positive numbers
With invalid entered name	28	74746466nn	Error message--> Accept with no characters only numbers
With invalid entered name	29	9737373%	Error message--> Accept with no special characters only numbers
With invalid entered name	30	1828288	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	1960 where salary 14000	accept
With invalid entered name	32	13448 where salary 5000	Error message--> Accept with the 14% value of salary
With invalid entered name	33	-45569 where salary 99000	Error message--> Accept with only positive numbers
With invalid entered name	34	1234lb where salary 9090	Error message--> Accept with no characters only numbers
With invalid entered name	35	500^ where salary 9000	Error message--> Accept with no special characters only numbers
With invalid entered name	30	7000000 where salary 90000	Error message--> Accept with only number<100000 And <1000
With valid entered name	31	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 21100898, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	accept

With invalid entered name	32	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 21100808, Nmuber of days 1000,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID doesn’t exists in database,Number of days more than 3 characters
With invalid entered name	33	Click create report button after entering the Employee name “Bardes Khalid”, Employee ID 211, Nmuber of days 12,Number of hours worked 133,salary 60000,VAT 8400	Reject Error message--> Employee ID less than 8 characters
Valid data or invalid data	34	Click cancel button	Accept(clear all entered data)

Test Requirements	Test case no	Values	Description
With Valid entered master card number	1	123456789123	Accept
With Invalid entered master card number	2	4.44323567829	Error message--> Accept with no decimal numbers
With Invalid entered master card number	3	678539	Error message--> Accept when it contains 12 characters
With Invalid entered master card number	4	-983135974	Error message--> Accept only positive numbers
With Invalid entered master card number	5	%123@^58	Error message--> Accept with no Special characters
With Invalid entered master card number	6	Mariam2341	Error message--> Accept with no Letters
With Invalid entered master card number	7	5 6 7 8 9 3 2	Error message--> Accept with no Spaces
With Valid entered master card expiration date	8	10/10	Accept
With invalid entered master card expiration date	9	12/2/6/8	Error message--> Accept when it contains only 5 characters
With invalid entered master card expiration date	10	0.6/7.2	Error message--> Accept with no decimal numbers
With invalid entered master card expiration date	11	-6/-11	Error message--> Accept only positive numbers
With invalid entered master card expiration date	12	2%&12	Error message--> Accept with no Specialcharacters
With invalid entered master card expiration date	13	5SfH7	Error message--> Accept with no

			Letters
With invalid enteredmaster card expiration date	14	5 7 8 9 0	Error message--> Accept with no Spaces
With Valid entered master card security code	15	5678392	Accept
With invalid entered master card security code	16	5678945632	Error message--> Accept when it contains only 7 characters
With invalid entered master card security code	17	6.6453327	Error message--> Accept with no decimal numbers
With invalid entered master card security code	18	-6799752	Error message--> Accept only positive numbers
With invalid entered master card security code	19	2@%$789	Error message--> Accept with no Special characters
With invalid entered master card security code	20	4D7G8hk8	Error message--> Accept with no Letters
With invalid entered master card security code	21	6 7 8 0 5 4 8	Error message--> Accept with no Spaces
valid data	22	Click pay button after entering master card number 123456789123, master card expiration date 10/10, master card security code 5678392	Accept
Invalid data	23	Click pay button after entering master card number 1233@34572$9786, master card expiration date 2$@1, master card security code 1267867	Reject Error message--> Master card number contains special characters,Master card expiration date contains special characters
Invalid data	24	Click pay button after entering master card number 78AB457f9786, master card expiration date 6 8 7 / 1 3, master card security code 5678953	Reject Error message--> Master card number contains Letters,Master card expiration date contains spaces

Test Req	**Test Case no**	**Component**	**Values**	**Description**
With valid code	1	***Course code***	CSE221	Accept
With invalid code	2	***Course code***	CSE221$	Error Message ---> Accept only letters and numbers
With invalid code	3	***Course Code***	CSE 221	Error Message --> No Space
With valid Grade	4	***Course Grade***	D/F	Accept

With valid Grade	5	***Course Grade***	D-/D	Accept
With invalid Grade	6	***Course Grade***	B13	Error Message --> No Numbers
With invalid Grade	7	***Course Grade***	15.344	Error Message--> Accept letters only
With invalid Grade	8	***Course Grade***	F@/C&	Error Message--> Accept letter only
With valid GPA/CGPA	9	***GPA/CGPA***	1.3/1	Accept
With invalid GPA/CGPA	10	***GPA/CGPA***	-2.3/-2	Error Message --> Only positive numbers
With invalid GPA/CGPA	11	***GPA/CGPA***	2B/B	Error Message--> Accept Numbers only
With invalid GPA/CGPA	12	***GPA/CGPA***	4.5/5	Error Message--> Accept < 4 only
With invalid GPA/CGPA	13	***GPA/CGPA***	<2>/1.	Error Message--> No character only numbers
With valid ID	14	***Student ID***	21100822	accept
With invalid ID	15	***Student ID***	20100bg	Error message--> Accept with no characters only numbers
With invalid ID	16	***Student ID***	2111000099469	Error message--> Accept with only 8 numbers
With invalid ID	17	***Student ID***	20087#	Error message--> Accept with no Special characters
With invalid ID	18	***Student ID***	-21100824	Error message--> Accept only positive numbers
With invalid ID	19	***Student ID***	3992456	Error message--> Student id not stored in database
			Action	**Description**
With valid data	20	***Create***	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGPA/GPA less than or equal 4	Accept
With invalid data	21	***Create***	Click Create button after entering the Student ID 21100824, Course Code CSE251, Course Grade A+, CGPA/GPA 3.6 where CGOA/GPA more than 4	Reject
Valid data or invalid data	22	Cancel	Click Cancel button	Accept (clear all entered data)

Test Requirement	Test Case Number	Values	Description
Valid value	1	d140	Accept and already stored in database
Invalid value	2	:09;	Error message: Contain characters
Invalid value	3	C-300	Error message: Can't be negative number

Invalid value	4	t087	Error message: Small letter
Invalid value	5	“blank”	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	6	“Yousef Ahmed Kamel”	Accept
Invalid value	7	Ahmed342	Error message: Contain number
Invalid value	8	Ahmed=-	Error message: Contain character
Invalid value	9	YousefAhmed	Error message: No spaces
Invalid value	10	“blank”	Error message: Enter value
Test Requirement	Test Case Number	Values	Description
Valid value	11	22100818	Accept
Invalid value	12	211[#86=	Error message: Contain character
Invalid value	13	211AbcD00	Error message: Contain letters
Invalid value	14	-22100818	Error message: Negative value
Invalid value	15	“blank”	Error message: Enter value
Invalid value	16	21100863818	Error message: Exceeded 8 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	17	99	Accept
Invalid value	18	)(96+	Error message: Contain character
Invalid value	19	cvER67	Error message: Contain letters
Invalid value	20	-90	Error message: Negative value
Invalid value	21	“blank”	Error message: Enter value
Invalid value	22	1862	Error message: Exceeded 3 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	23	1	Accept
Invalid value	24	[#=9	Error message: Contain character
Invalid value	25	AbcD67	Error message: Contain letters
Invalid value	26	-7	Error message: Negative value
Invalid value	27	“blank”	Error message: Enter value
Invalid value	28	9265	Error message: Exceeded 2 numbers
Test Requirement	Test Case Number	Values	Description
Valid value	29	Click edit button after entering Course ID D100,Student Name Yousef Ahmed	Accept

		Kamel ,Student ID 21100863 , Marks 99,Attendance 9	
Invalid value	30	Click edit button after entering Course ID D100,Student Name Yousef Ahmed Kamel ,Student ID 21100#*+ , Marks 99,Attendance 900	Error message: Invalid Student ID and Attendance
Invalid value	31	Click edit button after entering Course ID D8092,Student Name Yousef Ahmed Kamel ,Student ID 21100863 , Marks 99,Attendance 9	Error message: Invalid course ID
Test Requirement	Test Case Number	Values	Description
Valid or invalid	32	Click cancel button	Accept and clears all data

-

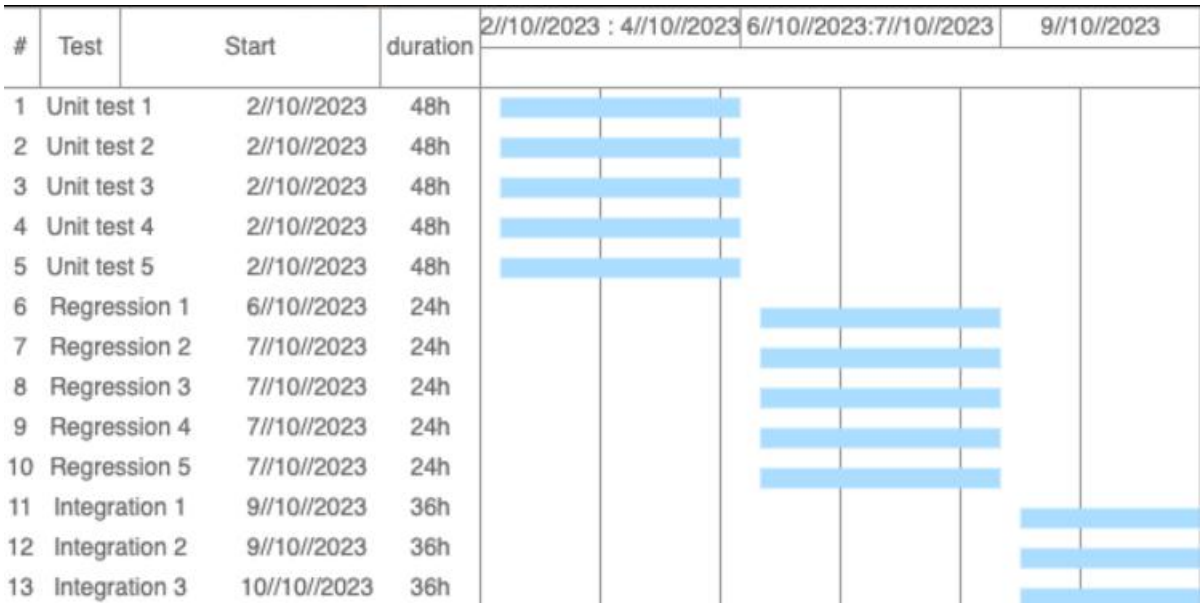
Test Requirement	*Test Case Number*	Component	*Values*	Description
Valid	1	User name	Hana Mohamed	Accepted and saved
invalid	2	User name	Hana Mohamed @	Error massage : cant contain any special characters "&*#$@'\?|<>,."
invalid	3	User name	Hana Mohamed 11	Error massage : cant contain any numbers"123456789"
invalid	4	User name	HanaMohamed	Error massage:There is no space between first name and last name
Test Requirement	*Test Case Number*	*Values*	Description	*Test Requirement*
With valid entered name	5	21100863	accept	With valid entered name
With invalid entered name	6	21100GHH	Error message--> Accept with no characters	With invalid entered name

			only numbers	
With invalid entered name	7	21118777777777777779	Error message--> Accept with only 8 numbers	With invalid entered name
With invalid entered name	8	21177#	Error message--> Accept with no Special characters	With invalid entered name
With valid entered name	9	01003471736	accept	With valid entered name
With invalid entered name	10	01003471736ooA	Error message--> Accept with only numbers	With invalid entered name
With invalid entered name	11	010034717	Error message--> Accept 12 numbers	With invalid entered name
With invalid entered name	12	010034717369999999	Error message--> More than 12 numbers	With invalid entered name
With invalid entered name	13	01003471736@#	Error message--> Accept with no special characters only numbers	With invalid entered name
With invalid entered name	14	01003471736.55	Error message--> Accept with no decimal numbers	With invalid entered name
Test Requirement	*Test Case Number*	*Values*	Description	*Test Requirement*
valid	15	**Click save** Name: Hana Mohamed ID:21100863 Phone : 01003471736	Accept and saved to database	valid
invalid	16	**Click save** Name: : Hana Mohamed #$ ID:21100SD	Error message-->	invalid

		Phone:01017230LLL	Accept with no characters only numbers	
invalid	17	**Click save** Name: Hana Mohamed 11 ID: 2110086333333 Phone:01014405	Error message--> Accept with only 8 numbers	invalid
invalid	18	**Click save** Name: HanaMohamed ID:211008@@ Phone : 01003471736.55	Error message--> Accept with no Special characters	invalid

Test	Duration	Dependability
Unit test 1	2 days	-------
Unit test 2	2 days	-------
Regression test 1	1 day	Dependable on unit test 1 & 2 should be done first
Unit test 3	2 days	-------
Unit test 4	2 days	-------
Regression test 2	1 day	Dependable on unit test 3 & 4 should be done first
Unit test 5	2 days	-------
Regression test 3	1 day	Dependable on unit test 1, 2, 3, 4 & 5 should be done first
Integration test 1	1.5 day	Dependable on Regression test 1 & 2 should be done first
Integration test 2	1.5 day	Dependable on Regression test 3 & 4 should be done first
Integration test 3	1.5 day	Dependable on Regression test 1, 2,3,4&5 should be done first

Gantt Chart:



#	Test	Start	duration
1	Unit test 1	2//10//2023	48h
2	Unit test 2	2//10//2023	48h
3	Unit test 3	2//10//2023	48h
4	Unit test 4	2//10//2023	48h
5	Unit test 5	2//10//2023	48h
6	Regression 1	6//10//2023	24h
7	Regression 2	7//10//2023	24h
8	Regression 3	7//10//2023	24h
9	Regression 4	7//10//2023	24h
10	Regression 5	7//10//2023	24h
11	Integration 1	9//10//2023	36h
12	Integration 2	9//10//2023	36h
13	Integration 3	10//10//2023	36h

- **Test cases for sub-systems:**

Test number	Test case objective	Test case description	Expected outcome
1	Check GUI components for given page	Open every page one by one and check GUI components functionality and presence	opened pages according to the page requested Successfully
2	Check login System	Enter User Name , password and click Login Button	A new page with all options should be open successfully
3	Check all information of the user stored in the database	Enter all user information	The user will receive a message that every detail about his information should be stored in the database
4	Check data generation of the server.	Enter the selected information set details and collect the new data when arrived.	Consistent data should be collected and reflected to the user either via graph or a table.
5	Check payment processes is done successfully	Click “pay” and fill all card information	A new page displays “ the payment is done succefully” should be open
6	Check store all new user info in the database	Add all new user info and click add	The system should store the new user in the database
7	Check if the system will	The system will check	The system will send a

	send to the student to the final semester report when admin creates and send it	every day if there is any created and failed to send	notification to the student to tell him that the final report of semester is sent.
8	Check linking between first page and course material page	Click to request course material button	A new page with all course material should be open
9	Check if the system send salary report created by the admin to employee	The system will check every day if there is any created and failed to send	The system will send a notification to the employee to tell him that the salary report is ready
10	Check if converting an information set to a selected format is working properly.	Create an information set with some mock data generated, and try to convert it to all supported formats.	Every information in the set should be presented nicely and clearly without any lacking information set, in all formats.
11	Check system provides a notification based on given parameters by the user or based on occurred errors.	Select an info set which in created by sending notification in given condition option and satisfy the given option the get notification	System should provide a notification to the user's application by using the server.
12	Check related data is retrieved from database	Select created information set and click "Live Data" button.	Incoming data should be provided via graph or table.
13	Check if the requested data is gathered and represented to the user correctly and nicely.	Try every activity page while trying also some data requests such as some information set details and data.	All data that is represented in the GUI should be exactly the same as the ones in the database, with correct places in the GUI.
14	Check that student who added a course is saved in the course students list	Database follows the course list until student is added	The system displays a succefully added course message.
15	Check that student who dropeed a course is saved in the course students list	Database follows the course list until student is added	The system displays a succefully dropped course message.
16	Check that the created student semester report is saved in database	The system checks for any unsaved reports	The system displays succefully added student final semester report and sent
17	Check that the created salary report is saved in database	The system checks foe any unsaved reports	The system displays succefully added salary report and sent
18	Check logout System	The user will click "log out" button	A registration or log in page should be open

- **Sub-systems with required test cases:**

System	Test cases required

Controller management	1,3,4,6,7,8,9,12,13,20
Courses management	2,14,15
Database management	3,6,11,12,14,15,16,17
System management	2,6,9,16,17
Courses in progress management	2,8
Payment option management	2,5