CS390 Software Engineering – Fall 2024
Assignment 1

Ahmed Abdelbary 2221191298
Lamiaa Abushara 2211150120
Seba Alzahir 2201114862

Table of Contents

Individual Contributions

Task	Ahmed	Lamiaa	Seba
Introduction	40%	30%	30%
Customer Statement of requirements	30%	30%	40%
Glossary	25%	25%	50%
System requirments	40%	30%	30%
Functional requirements specification	50%	25%	25%
User interface specification	30%	35%	35%
Risk management	30%	40%	30%
Project plan	25%	50%	25%
References	30%	35%	35%

1. Introduction

The Computer Science Assistance Request and Credit System (CS ARCS) has this document as its official system specification, a software program designed to enhance classroom interactions within the Computer Science department. CS ARCS encourages systematic support and guidance by allowing junior students to ask for help on difficult academic subjects and senior students to offer this support in a supervised, university approved setting.

This ARCS system of CS aims at facilitating the process of peer-to-peer catered around the junior student academic needs seeking assistance and the senior students acting as mentors. The faculty members overseeing this interaction control the compliance of the coursework with the departmental standards and offer additional points for those seniors who manage to carry it out. The system features are designed to enhance task allocation, ensure fairness and provide a pleasant interface for the user.

This document also provides the project agenda, graphical user interface, risk management plan, detailed non-functional and functional requirements as well as the requirements of the customer. There are plans for the CS ARCS system to offer a robust, cost-effective and easy to navigate system designed to encourage learning and interaction within the Computer Science Department. The requirements and specifications presented here are expected to serve as a foundation for its development and implementation.

2. Customer Statement of Requirements

Customer Statement of Requirements (CSR)

Overview

As a client representing the Computer Science Department, we aim to implement a system that supports well-structured peer-to-peer learning. Junior students can ask for assistance with subjects that is difficult for them and senior students can provide guidance in a supervised environment. This program should be advantageous for all stakeholders involved(juniors, seniors, and faculty) and integrate into our CS department's operations.

Objectives

1. **Facilitate Academic Support**: We are keen to provide junior students with a user-friendly interface, where they can request for assistance in the specific difficult courses.
2. **Encourage Senior Mentorship**: The seniors should not only be the passive recipients of knowledge, but instead be encouraged to serve as mentors to the juniors and in the process pass on their knowledge. By awarding additional credits for tutors, we envision creating a positive and nurturing environment within our department.
3. **Ensure Faculty Supervision:** Faculty supervision of the interaction is essential so that the support is in accordance with departmental and academic integrity policies.
4. **Create a User-Friendly System**: The system should be user friendly to enable the students and even the faculty members to work on it without any difficulties. We envision a design that is simple with few instructions, easy task allocation and a straightforward follow up of the sessions.

Core Features

1. **Request Assistance Portal**:
   Junior students should be able to submit with the chosen subject, topic and the kind of help that they seek.
   Browse file / request specific link section would be a welcome addition as optional.
2. **Mentorship Assignment**:
   Senior students who are available and qualified in particular subjects should receive notifications of relevant assistance requests.
   Assigning should be proportionate so that each senior takes on a certain amount of mentorships, promoting equity in the assigning of responsibilities.
3. **Mentorship Dashboard**:
   Senior mentors should have to access a dashboard which allows them to see all the assignments that are active as well as the history of the mentors and the juniors under them.
   Further provisions, like allowing people being mentored by juniors to rate the help they received and allowing seniors to have a courtesy of expressing their thoughts regarding the help they gave, would enrich the system.
4. **Faculty Supervision Console**:
   Faculty members should be able to oversee all ongoing sessions, monitor conversation history, and view feedback to ensure quality control.
   Faculty should have the ability to award credit points to seniors based on participation and feedback scores.
5. **Progress Tracking and Reporting**:
   Requests and sessions progress should also be viewed both by juniors and seniors.
   Automated reports on mentorship hours and assistance provided should be accessible to faculty for assessment purposes.

3. Glossary

CS ARCS: Computer Science Assistance Request and Credit System – a platform that facilitates peer-to-peer mentoring within the Computer Science department.

Junior Student: A student in an earlier year of study (typically first or second year) who requests help on challenging subjects or topics.

Senior Student: A student of an betted year such as third year or fourth year whose role is to assist the junior students in a controlled environment and guidance too.

Faculty Member: a supervising teacher in charge of the mentoring session's quality output and adherence to the department's regulations.

Request Assistance Portal The application where junior students create and submit help requests indicating the subject and the form of help needed.

Mentorship Assignment: This is the process whereby junior students' requests for assistance are brought to assist eligible senior students according to their expertise and availability.

Mentorship Dashboard: An interface from which senior mentors can see the requests they have been assigned, monitor the progress of their mentorship activities and communicate with junior students.

Faculty Supervision Console: The interface used by faculty to oversee mentoring sessions, monitor progress, and allocate credits to senior mentors based on performance.

Credit Points: Extra points or marks given to final year students for properly playing the role of the mentor.

Task Allocation: A system characteristic whereby requests are fairly shared among the qualified mentors to avoid injustice.

Progress Tracking: A system characteristic whereby the users can see the progress and result of mentorship sessions.

User Interface (UI): The part of the system that users interact with, designed to be intuitive and user-friendly.

Scalability: The ability of the system to handle increasing number of users and requests, especially in peak periods.

Security: Measures to protect user data and ensure access control within the system.

Responsiveness: The adaptability of the system across various devices and screen sizes, allowing for easy access on desktops, tablets, and smartphones.

Reliability: The system's ability to function without disruptions, providing consistent access to assistance sessions.

Session Feedback: A feature allowing junior students to provide feedback on the assistance they received, which helps faculty assess mentor performance.

Automated Reports: Reports generated by the system automatically that are responsible for providing summaries of mentorship hours and assistance activities for faculty review.

User: Any individual accessing the system, including junior students, senior mentors, and faculty members.

Peer-to-Peer Learning: A learning approach where students learn from each other through mentorship and collaboration.

4. System Requirements

4.1. Functional Requirements

ID	Priority weight	Requirement description
F-01	5	System should authenticate users and differentiate roles. (Log in/sign in)
F-02	5	Junior students can create and manage task requests. Senior students should be able to browse and accept tasks.
F-03	4	System must verify senior students' tasks align with courses.

F-04	3	Track task status, documents exchanged, and scheduling details.
F-05	5	Faculty can review tasks and grant extra credits if standards met.
F-06	2	Notify users about task postings, acceptances, and completions.
F-07	4	Implement task prioritization and fair distribution mechanism
F-08	2	Enable feedback and rating system for tutoring sessions.
F-09	3	Allow faculty to manually override system decisions on credits
F-10	2	Enable document sharing and collaboration tools within tasks.
F-11	2	Provide a search function for users to find specific tasks.
F-12	3	Include a calendar system for scheduling and reminders.

4.2. Non-Functional Requirements

ID	Priority Weight	Requirement Description
NF-01	5	Ensure system availability 24/7 with minimal downtime.
NF-02	4	Interface should be user-friendly and tailored to each user role.
NF-03	5	System should be responsive on various devices including mobile.
NF-04	5	Secure user data and prevent unauthorized access.

NF-05	4	Ensure data integrity and prevent loss or corruption.
NF-06	4	Comply with accessibility standards to support all users.
NF-07	2	Integrate smoothly with existing university systems.
NF-08	4	Designed for easy maintenance and future updates.
NF-09	3	System should maintain a log of all activities for auditing.
NF-10	3	System updates should be deployable without downtime.
NF-11	3	Provide multilingual support for diverse student bodies.
NF-12	2	Ensure system's user interface adheres to the latest UI trends.

5. Functional Requirements Specification

5.1. Stakeholders

Junior Students: These are mainly users who create task requests in the system in order to get tutorial help on certain topics in computer science. Their intention, most of the time, is to seek help for areas that are difficult for them.
Senior Students: These are users, whose enrolled courses allow them to accept task requests from juniors and provide tutorial help. Their aim is to help the juniors and earn some points for the help provided.

Faculty Members: Users whose duty is to oversee the quality of the assistance provided, ensuring it meets the standards of the department, as well as giving scores to the senior students for the tasks done.

University Library System: An off the shelf system that tracks access to rooms for junior and senior students when they are attending classes as conclusive attendance records.

5.2. Actors

Junior Student: Commences requests for help regarding a given task as well as the particular subject and type of assistance required.

Senior Student: Attends to and takes task requests within his or her area of expertise, engages in tutoring, and is awarded bonus marks.

Faculty Member: Helps in the fulfilling of the tasks set, lawfully assesses the feedback of the sessions held, and assigns senior students with additional credit points.

Library Room Access System: A database system that logs students every time they enter controlled areas with the aim of ensuring that the students in question carried out the mentorship within approved university premises.

5.3. Use Cases

5.3.1. Use Case Diagram

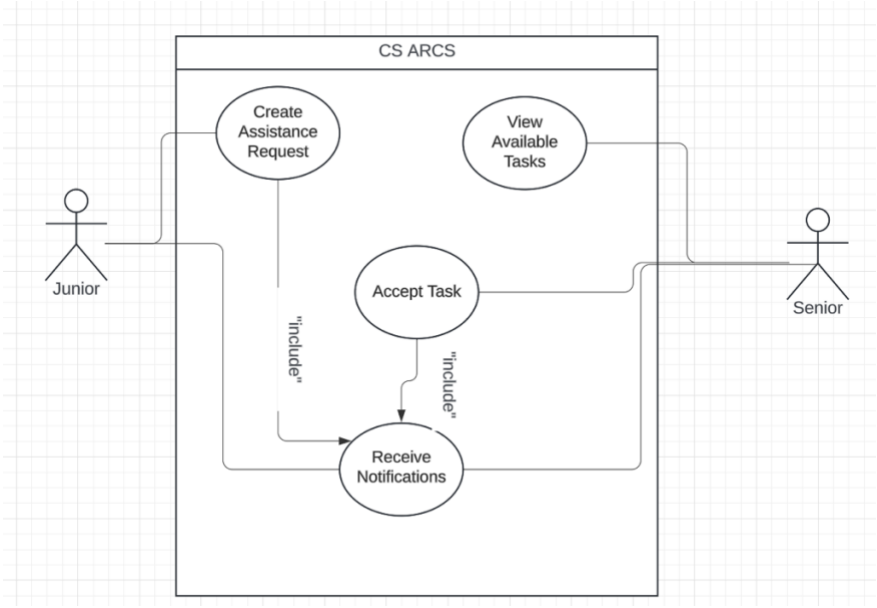

Fig. (1a) : Student Task Management Diagram:

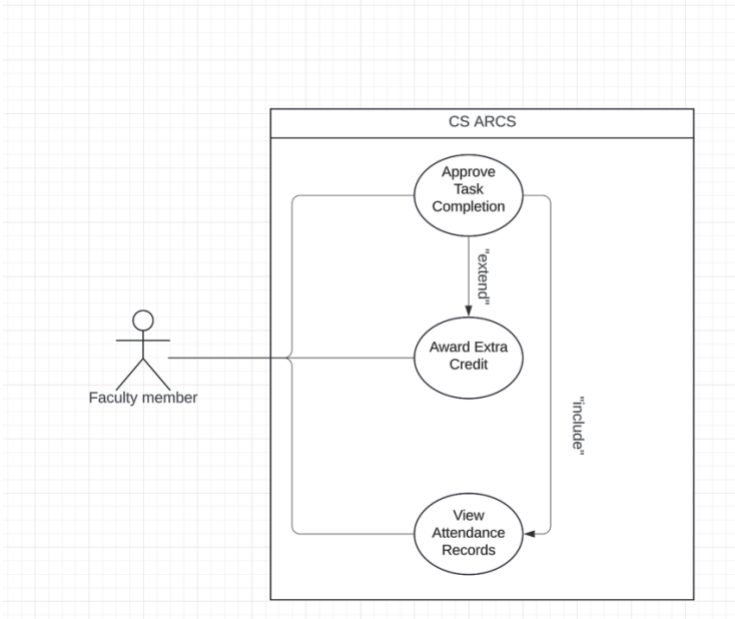

Fig. (1b) : Task Approval and Credit Allocation Diagram

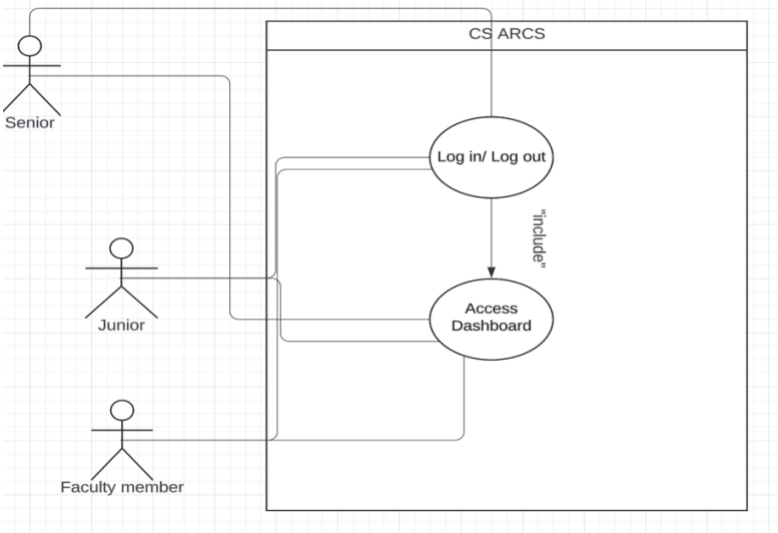

Fig. (1c) : User Authentication and Access Control Diagram

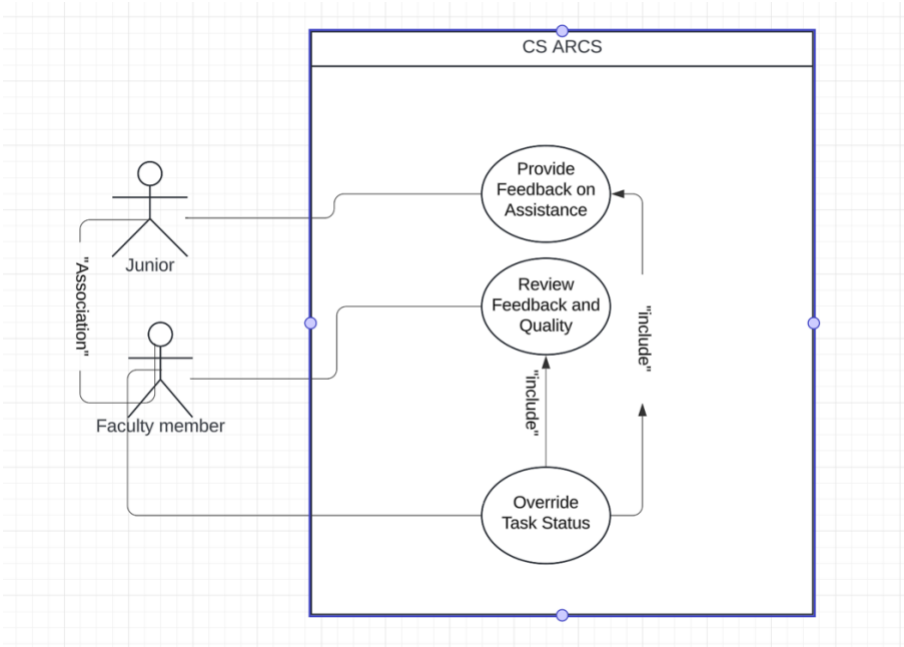

Fig. (1d) : Feedback and Quality Assurance Diagram

5.3.2. Use Case Summaries

ID	Name	Description
UC-1	Create Assistance Request	The Junior Student submits a request for tutoring, specifying the subject area and the type of guidance needed. This request is then posted to the system where eligible Senior Students can view and accept it.
UC-2	View Available Tasks	The Senior Student browses the list of open requests submitted by Junior Students. The system filters tasks based on the Senior Student's enrolled courses to ensure only relevant tasks are displayed.

UC-3	Accept Task	The Senior Student selects and accepts a task that aligns with their expertise. Once accepted, the task is assigned to the Senior Student, and notifications are sent to both the Junior Student and Faculty Member.
UC-4	Approve Task Completion	The Faculty Member reviews the completed task and assesses its quality. Based on the assessment, the faculty member approves or rejects the task completion and awards extra credit if the standards are met.
UC-5	Verify Room Access	This system use case records the room entry and exit times of both Junior and Senior students in designated library rooms, providing proof of attendance for the tutoring session.
UC-6	Provide Feedback on Assistance	After receiving tutoring, the Junior Student provides feedback on the session. This feedback is submitted to the Faculty Member for quality control purposes.
UC-7	Award Extra Credit	Based on the completed task and feedback, the Faculty Member awards extra credit to the Senior Student. This credit is linked to the student's coursework and incentivizes high-quality tutoring.
UC-8	Receive Notifications	The system sends notifications to Junior and Senior students when key events occur, such as when a task is posted, accepted, or

		completed. This ensures that users stay updated on task progress.
UC-9	Log In	The user (Junior, Senior, or Faculty) logs into the system, authenticating their role to gain access to relevant features. Each role has a unique dashboard, enabling specific functionalities aligned with their needs.
UC-10	Monitor Task Progress	The Faculty Member can view the status of all active tasks, including tutoring session schedules and progress updates, to ensure that tasks are on track and comply with departmental standards.
UC-11	Access Dashboard	users (Junior Student, Senior Student, Faculty Member) access a dashboard tailored to their role, where they can see relevant tasks, updates, or approvals.
UC-12	Review Feedback and Quality	Faculty Members review feedback submitted by Junior Students and assess the quality of the assistance provided by Senior Students. This use case supports quality control for awarded credits.

5.3.3. Traceability Matrix

	UC-1	UC-2	UC-3	UC-4	UC-5	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12
F-01										X			
F-02	X	X	X									X	
F-03			X										
F-04				X							X		
F-05				X				X					
F-06									X				
F-07		X									X		
F-08							X						X
F-09				X				X					X
F-10				X									
F-11		X											
F-12												X	

5.3.4. Fully dressed use case diagrams

UC-1: Create Assistance Request

Initiating Actor: Junior Student

Actor Goal: To submit a request for assistance on a specific course topic.

Stimulus: The Junior Student logs into the system and navigates to create a new assistance request.

Precondition: The Junior Student is authenticated and has access to the "Request Assistance" feature.

Postcondition: A new assistance request is created and visible to eligible Senior Students.

Flow of Events (Main Success Scenario):

1. Junior Student logs into the CS ARCS system.
2. Junior Student navigates to the "Request Assistance" page.
3. Junior Student specifies the course, topic, and type of help required.
4. Junior Student optionally uploads any relevant files or links.
5. Junior Student submits the assistance request.
6. The system saves the request and notifies eligible Senior Students.

Alternative Scenario(s):

Missing Information: If the Junior Student leaves required fields blank, the system prompts them to fill in all mandatory details before submission.

UC-2: View Available Tasks

Initiating Actor: Senior Student

Actor Goal: To browse available assistance requests and choose tasks.

Stimulus: The Senior Student logs into the system and views the list of available tasks.

Precondition: The Senior Student is authenticated and has access to browse tasks.

Postcondition: The Senior Student has reviewed available tasks and can accept an appropriate one.

Flow of Events (Main Success Scenario):

1. Senior Student logs into the CS ARCS system.
2. Senior Student navigates to the "Available Tasks" section.
3. The system displays tasks that match the Senior Student’s courses.
4. Senior Student reviews the list of tasks and selects a suitable one for assistance.

Alternative Scenario(s):

No Matching Tasks: If no tasks are available that match the Senior Student's courses, the system displays a message indicating no suitable tasks.

UC-3: Accept Task

Initiating Actor: Senior Student

Actor Goal: To accept a task and commit to providing assistance.

Stimulus: The Senior Student selects a task from the available tasks list.

Precondition: The Senior Student is authenticated and has permission to accept the task.

Postcondition: The task is assigned to the Senior Student, and the Junior Student is notified.

Flow of Events (Main Success Scenario):

1. Senior Student selects a task from the "Available Tasks" section.
2. Senior Student clicks on "Accept Task."
3. The system verifies the Senior Student's enrollment in the relevant course.
4. The system assigns the task to the Senior Student.
5. The system notifies the Junior Student that their task has been accepted.

Alternative Scenario(s):

Task Already Taken: If another Senior Student has already accepted the task, the system displays an error and removes it from the available list.

UC-4: Approve Task Completion

Initiating Actor: Faculty Member

Actor Goal: To review completed tasks and award extra credit if standards are met.

Stimulus: Faculty Member logs into the system and reviews a completed task.

Precondition: The task is marked as "Completed" by the Senior Student and ready for faculty review.

Postcondition: The task is approved, and extra credit is awarded to the Senior Student.

Flow of Events (Main Success Scenario):

1. Faculty Member logs into the CS ARCS system.
2. Faculty Member navigates to the "Review Completed Tasks" section.
3. Faculty Member selects a task for review.
4. Faculty Member reviews the task details, session logs, and feedback.
5. If the task meets standards, Faculty Member approves the task and awards extra credit.
6. The system updates the task status to "Approved" and notifies the Senior Student.

Alternative Scenario(s):

Does Not Meet Standards: If the task does not meet standards, Faculty Member marks it as "Requires Improvement," and the system notifies the Senior Student.

UC-5: Verify Room Access

Initiating Actor: Library Room Access System

Actor Goal: To verify that tutoring sessions occur in designated library rooms.

Stimulus: Junior and Senior students access the library room for a tutoring session.

Precondition: Junior and Senior students are registered for a tutoring session.

Postcondition: Attendance is recorded in the system for both students.

Flow of Events (Main Success Scenario):

1. Junior and Senior Students arrive at the designated library room.
2. Each student scans their ID upon entering the room.
3. The library system records the entry time and confirms both students' presence.

4. The system saves attendance data as proof of participation for faculty review.

Alternative Scenario(s):

No Match Found: If either student does not scan in, the system logs incomplete attendance for the session.

UC-6: Provide Feedback on Assistance

Initiating Actor: Junior Student

Actor Goal: To provide feedback on the assistance received from the Senior Student.

Stimulus: The Junior Student completes a session with the Senior Student.

Precondition: The assistance session has concluded.

Postcondition: Feedback is recorded and available for faculty review.

Flow of Events (Main Success Scenario):

1. Junior Student logs into the CS ARCS system.
2. Junior Student navigates to "My Tasks" and selects the completed session.
3. Junior Student rates the assistance and leaves additional comments.
4. The system records the feedback and notifies the Faculty Member.

Alternative Scenario(s):

Skip Feedback: If the Junior Student skips feedback, the system sends a reminder after a set time.

UC-7: Award Extra Credit

Initiating Actor: Faculty Member

Actor Goal: To award extra credit based on task completion and quality.

Stimulus: Faculty Member reviews completed tasks with satisfactory ratings.

Precondition: Task has been reviewed and marked as approved.

Postcondition: Extra credit is added to the Senior Student’s record.

Flow of Events (Main Success Scenario):

1. Faculty Member reviews the approved task.
2. Faculty Member assigns extra credit points.
3. The system updates the Senior Student's record with extra credit awarded.
4. The Senior Student is notified of the extra credit.

Alternative Scenario(s):

Credit Denied: If feedback is unsatisfactory, Faculty Member denies extra credit, and the system notifies the Senior Student.

UC-8: Receive Notifications

Initiating Actor: Junior Student, Senior Student

Actor Goal: To stay updated on task status changes.

Stimulus: A relevant event occurs (task posting, acceptance, or completion).

Precondition: The student has an active profile in the system.

Postcondition: Notification is sent to the relevant student(s).

Flow of Events (Main Success Scenario):

1. System detects a task status change.
2. System sends a notification to the relevant students (Junior/Senior).
3. Students receive the notification on their dashboard or via email (if enabled).

Alternative Scenario(s):

Notification Disabled: If notifications are disabled, students can check status manually on their dashboard**.**

UC-9: Log In

Initiating Actor: Junior Student, Senior Student, Faculty Member

Actor Goal: To securely access the system and view role-specific functionalities.

Stimulus: The user opens the CS ARCS login page and enters their credentials.

Precondition: The user has a valid account in the system.

Postcondition: The user is logged in and directed to their specific dashboard (Junior, Senior, or Faculty).

Flow of Events (Main Success Scenario):

1. User opens the CS ARCS login page.
2. User enters their username and password.
3. The system verifies the credentials.
4. If credentials are correct, the system logs in the user and directs them to their specific dashboard.
5. The user gains access to functionalities specific to their role.

Alternative Scenario(s):

Invalid Credentials: If the entered credentials are incorrect, the system displays an error message and prompts the user to try again.

Forgot Password: If the user cannot remember their password, they can initiate a password reset.

UC-10: Monitor Task Progress

Initiating Actor: Faculty Member

Actor Goal: To track the progress and status of all active tasks.

Stimulus: Faculty Member logs into the system and navigates to the "Monitor Task Progress" section.

Precondition: Faculty Member is authenticated and logged in.

Postcondition: The Faculty Member has a clear view of all ongoing tasks and their current statuses.

Flow of Events (Main Success Scenario):

1. Faculty Member logs into the CS ARCS system.

2. Faculty Member navigates to "Monitor Task Progress."
3. The system displays a list of all active tasks, with information on task status, assigned students, and scheduling details.
4. Faculty Member reviews task statuses and can drill down into specific task details if needed.

Alternative Scenario(s):

No Active Tasks: If there are no active tasks, the system displays a message indicating that all tasks are complete or awaiting approval.

UC-11: Access Dashboard

Initiating Actor: Junior Student, Senior Student, Faculty Member

Actor Goal: To access a personalized dashboard with relevant options based on their role.

Stimulus: The user logs into the system and is directed to their dashboard.

Precondition: User is successfully logged into the system.

Postcondition: The user sees role-specific functionalities and options on their dashboard.

Flow of Events (Main Success Scenario):

1. User logs into the CS ARCS system.
2. The system identifies the user's role (Junior Student, Senior Student, or Faculty Member).
3. The system directs the user to their role-specific dashboard.
4. The dashboard displays relevant options such as task requests (Junior), available tasks (Senior), or task review (Faculty).

Alternative Scenario(s):

Role-Switching (Admin Only): If there's an administrative role with access to multiple dashboards, the user can switch between views

UC-12: Review Feedback and Quality

Initiating Actor: Faculty Member

Actor Goal: To review feedback submitted by Junior Students on the assistance provided and assess the quality of completed tasks.

Stimulus: The Faculty Member navigates to the task feedback section.

Precondition: Completed tasks have feedback submitted by Junior Students.

Postcondition: Feedback is reviewed, and task quality is assessed.

Flow of Events (Main Success Scenario):

1. Faculty Member logs into the CS ARCS system.
2. Faculty Member navigates to the "Review Feedback and Quality" section.
3. Faculty Member selects a task with completed feedback.
4. Faculty Member reviews the Junior Student's feedback and any notes on the assistance session.
5. Faculty Member assesses the quality and decides if extra credit should be awarded or if further follow-up is needed**.**

Alternative Scenario(s):

Negative Feedback: If feedback is negative, the Faculty Member may mark the task as "Requires Improvement" and request a follow-up from the Senior Student.

5.4. System sequence diagrams

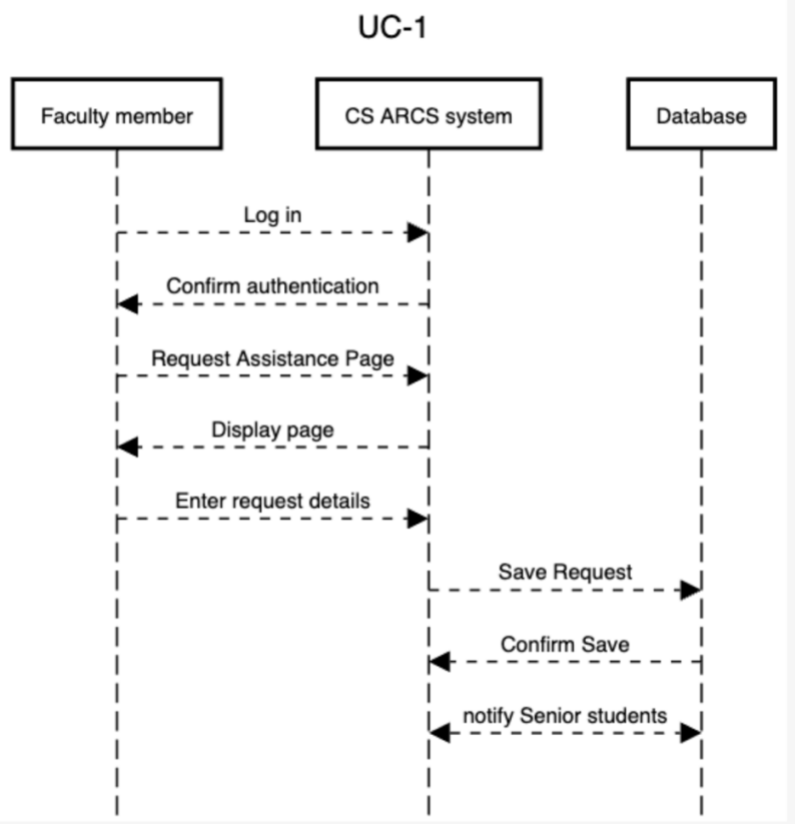

Fig. (2a) : sequence diagram for UC-1

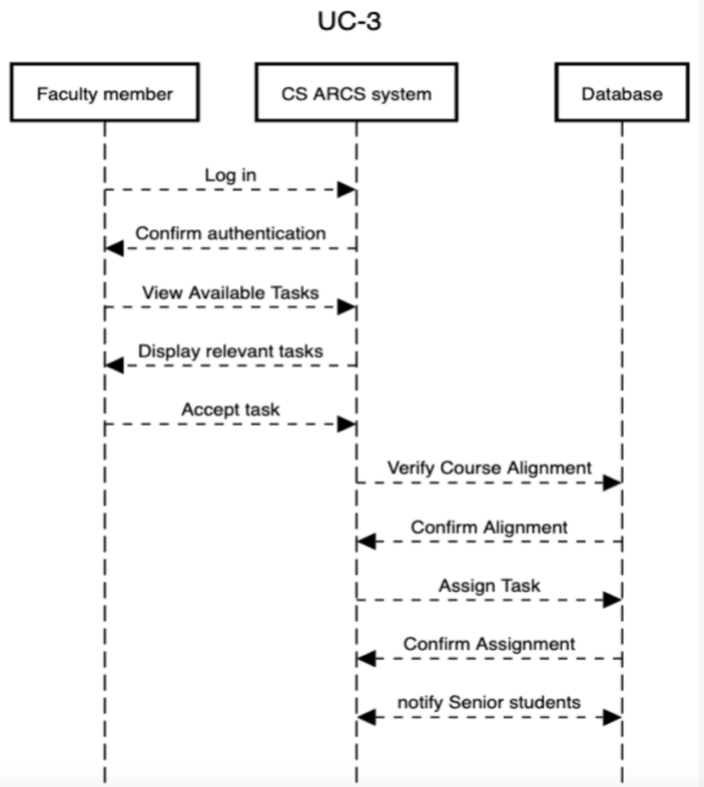

Fig. (2b) : sequence diagram for UC-3

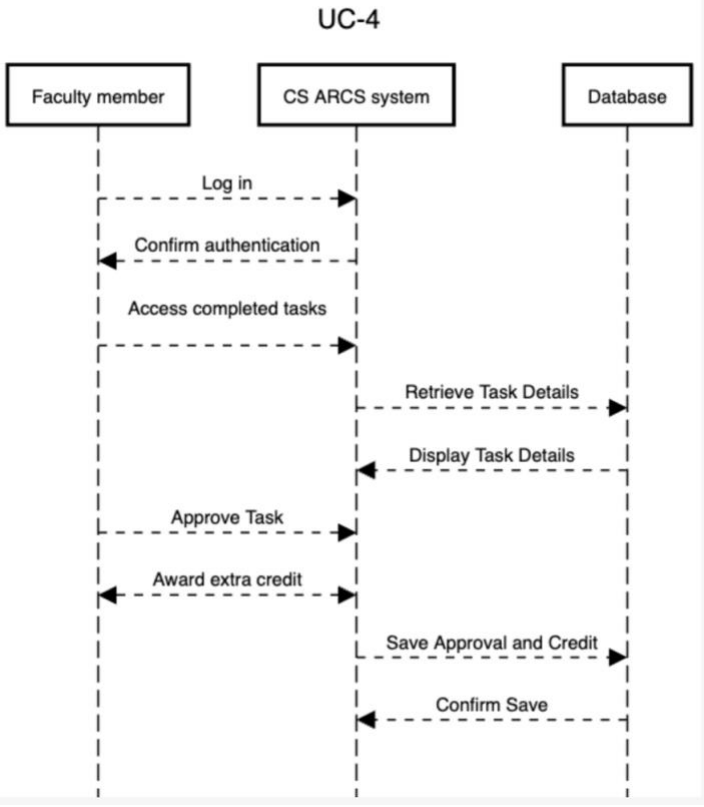

Fig. (2c) : sequence diagram for UC-4

6. User Interface Specification

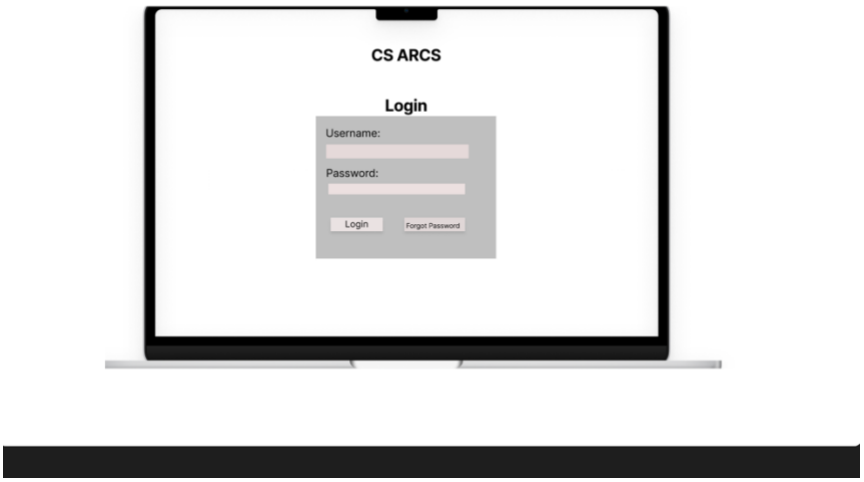

Fig. (3a) : main login page on a PC/Laptop (UC-9)

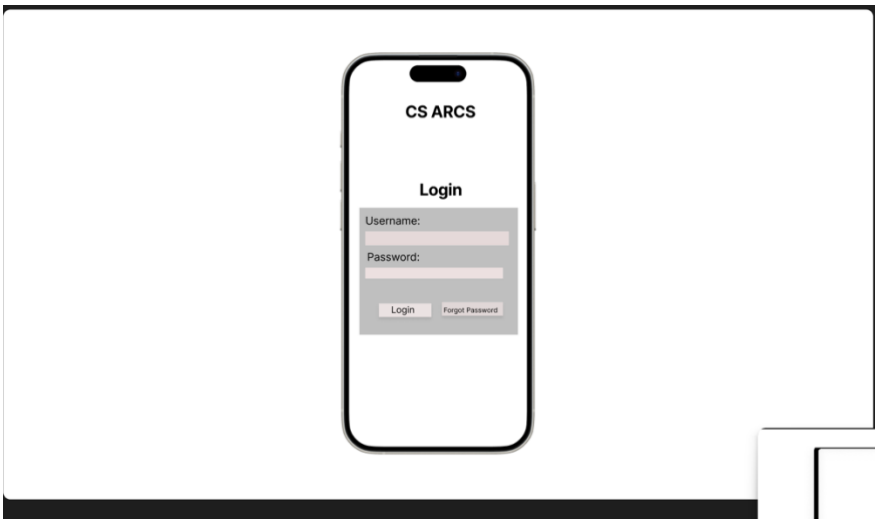

Fig. (3b) : main login page on a smartphone (UC-9)

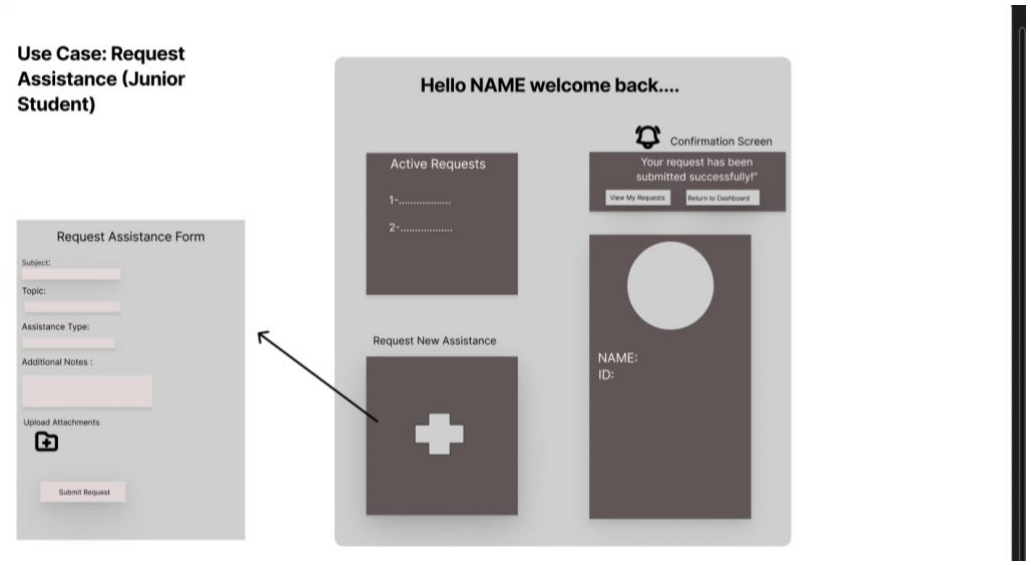

Fig. (3c) : Dashboard for Junior students (UC-1& UC-11)

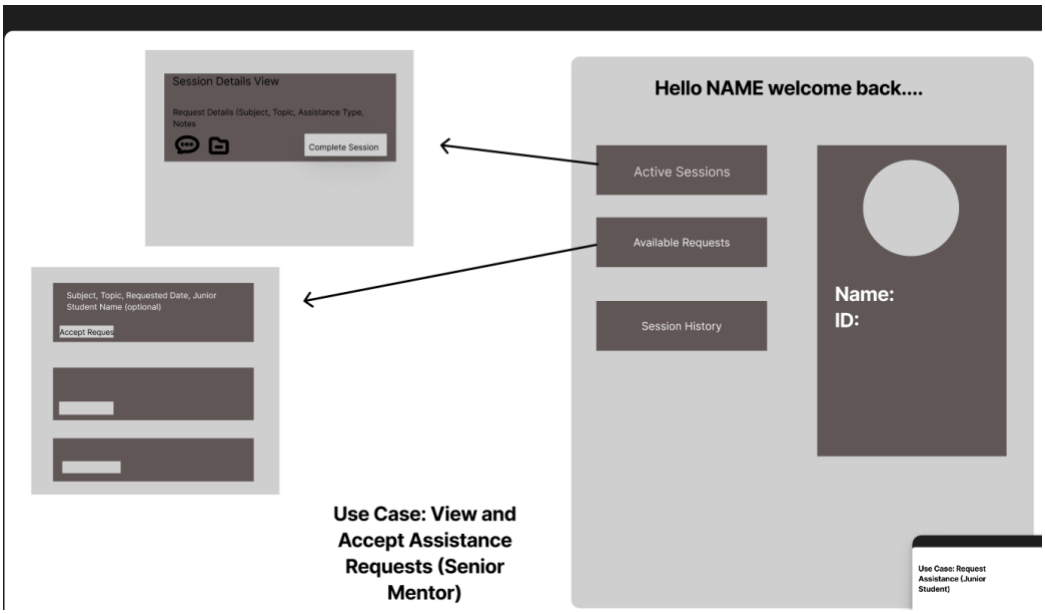

Fig. (3d) : Dashboard for Senior students (UC-2 & UC-3 & UC-11)

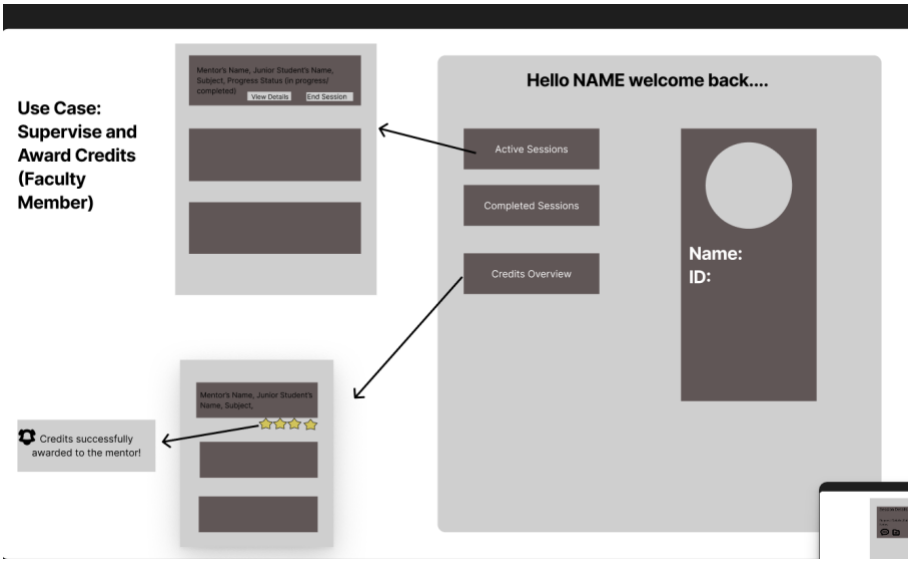

Fig. (3e) : Dashboard for Faculty members (UC-4 &UC-11 & UC-12)

7. Risk Management

Risk	Probability	Effects
Security risk due to unauthorized access to sensitive users' information.	4	Bad reputation, increased costs, and identity theft.
system crashes	3	it can disrupt the tutoring platform
Inadequate risk management	2	can lead to project failure
Scope Creep: This condition happens when the project’s features and functionalities keep expanding beyond the original plan.	5	budget overruns, missed deadlines, wasted time.
Low stockholder engagement	3	misalignment with stakeholder's needs and expectations.
low code quality	2	Poor user experience
Inaccurate deadline estimations	3	Late project delivery

8. Project Plan

Week	Milestones	Lamiaa	Ahmed	Seba
1	Research	Research and understand project		
1	Gathering requirements	documenting, and understanding the needs and goals of a project		
2	System design	Design UI	Create system architecture and diagrams	Design data flow
3-5	Development phase 1	Develop UI components	Develop task management functionality	Develop notification system
6-8	Development phase 2	Develop feedback system	Integrate all components	
9	Testing phase 1	Write test case	Implement test cases	Testing and fixing bugs

9. References

List all references used to create this report using APA format.

Books and Articles:

Russ Miles and Kim Hamilton, 2006. Learning UML 2.0 , O'Reilly Media.

Websites and online tools:

Lucidchart. (n.d.). *How to create sequence diagrams*.
https://www.lucidchart.com

UML Use Case Diagrams.
https://www.uml-diagrams.org/use-case-diagrams.html

Sequence diagram.
https://en.wikipedia.org/wiki/Sequence_diagram

Software Development Risks: A Detailed Guide
https://eluminoustechnologies.com/blog/software-development-risks/