Faculty of Engineering and Technology

Computer Science Department

COMP433 - Group Assignment Phase 4

EduSmart

Students Name	Student Number
Sarah Hassouneh	**1210068**
Dana Hafithah	**1211234**
Doaa Hatu	**1211088**
Leen Daraghmeh	**1210904**
Diana Naseer	**1210363**

Instructor : Dr. Samer Zein

Section: 1

Date : 15 Jan 2025

Introduction

In the past few years, and with the increase in the number of students and teachers, the concept of **“Modern Schools”** saw the light. With this revolution, the need for a **portal** that can achieve efficient management for the school system was a must. The process of managing core functions such as registration, submissions, schedules, payments, and any other communication is very hard to do manually, especially as the institution continues to grow. Many issues may occur, such as the delays in communication, problems with financial management and payments, difficulties in managing the large number of students and staff, and other difficulties in monitoring the academic process. There is obviously an urgent need for an automated system that can manage all the processes, ensuring efficiency, accuracy, and improved communication among all stakeholders.

The current school system faces many challenges that hinder day-to-day operations. One of the most pressing issues is the **Registration Process**. While it could be done online easily, the registration by manually handling several papers and information can cause difficulties and confusion to the secretary.
Another major challenge is **Schedule Management**. Coordinating multiple classes for teachers and students without a centralized system can result in conflicts, and make it harder to access up-to-date information.

Communication is another issue that needs to be solved. There must be a way for teachers to send announcements, updates, messages to students and parents. Teachers sometimes need to provide students feedback, and to communicate with parents to keep them updated on their children's progress. In addition to the communication between managers and parents for specific academic issues and payments.
As discussed before, the **Payment Process** needs to be managed perfectly as well, Including fees, payments due, and payroll for staff. The manual handling may cause serious risks, this shows how important the existence of a payment system for the accounting department is.

Assignment submission causes another challenge for teachers, since they need an effective way to send assignments and to keep track of the completion of student’s tasks. Doing this process manually can cause confusion in deadlines or misses in some tasks.

Since 2020, COVID-19 has changed the way schools operate, accelerating the need for digital solutions that support remote learning. The current school system was not equipped to handle this demand. Teachers and students struggled to stay in contact, and learn effectively because of the lack of an integrated platform. The proposed solution seeks to address these challenges by providing an excellent and complete platform that supports both remote and in-person learning, allowing for real-time communication, assignment submissions, secure payments, and scheduling. Therefore, the system will be beneficial for emergencies and for every day school operations.

Our solution is to create a **Smart system** designed to transform all daily operations to suit modern school's needs. The system will help teachers, managers, students, parents, by automating tasks such as student registration, assignment management, accounting, communication all in one platform to keep it simple and efficient.

this user-friendly platform is made to adapt every stakeholder including managers, teachers, students, and parents, administrators will easily manage registrations and schedules, streamlining their tasks, teachers can focus on creative teaching methods instead of being overwhelmed by paperwork, students will learn from a more organized learning environment, while parents will always have a full insight into their child's progress.

Nowadays modern schools are not just looking for a functional system to serve them but an innovative one, and that's what sets us apart, we focus on flexibility, adaptability, and creativity , a solution that aligns perfectly with modern school's unique requirements, the platform is designed to grow with you, we are committed to offer customizations and improvements on the system as your school keeps developing by providing regular updates to ensure that it always meets the school's evolving needs. Furthermore, we prioritize security and privacy ensuring all your data is always safe. By choosing us, you are choosing excellence, reliability, and peace of mind. Our commitment is a long-term partnership to help you continuously improve, innovate, and succeed.

Proposed Solution

The modern school system is aimed to help to automate the school's tasks. It focuses on enhancing the learning experience for the students by motivating the students to participate more and engage effectively with the learning materials. The system aims to do that while maintaining a good management environment for the teachers, principal, parents and the school staff. It offers solutions using smart features to improve seamless communication between the school components. The system will be built as a web-based system while integrating some mobile features.

Proposed Features:

<u>Communication Features:</u>

1. The system will allow the students to send messages to their teachers, or reply to sent messages. The teachers will be able to send messages to a section, individual student, principal and the registration staff.
2. The system will provide a **"Raise a question" feature**, where a student can post a question to a forum that includes the teacher and the students enrolled in the subject. The question can be answered by the participants thus engaging the students.
3. The school system will communicate with the parents about their children's performance, grades, attendance and warnings, through a notification system.
4. The school system will enable the parents to send messages to their children's teachers and allow them to assign meetings with school teachers. **It will have a feature to auto-generate online meeting links.**
5. The system will provide teachers with a shared board where they can communicate and upload learning materials.
6. The system will provide an **exam builder feature**, where a teacher can easily add different types of questions (multiple choice, fill-in-blanks, theoretical questions, etc..). There will be a score for the exam based on the school's metrics (the length and type of questions) that will help the teacher build a balanced exam. The teacher can post this as an online quiz or print it.
7. For online quizzes there will be a feature to auto-grade where possible.

Task-distribution Features:

8. The system will help teachers in **load-balancing**, where they can assign quizzes and assignments on days where the students don't have any quizzes or assignments. This helps in solving any contradictions and helps students stay balanced.
9. The system will build a tailored calendar to each student, where the assignments' deadlines, quizzes are shown in real-time. The system will also provide reminders for deadlines for any uncompleted tasks and upcoming exams.
10. The system will provide a feature to **automate a weekly plan**. Teachers will provide the weekly materials and tasks for their subject and the system will **collect these from different teachers** by the end of the week to build a plan. The plan will be approved by the principal and then it could be viewed by students and parents.

Accounting Features:

11. The system will provide parents with **online payment methods**. It will also track their previous payments and will issue reminders for the payments.
12. The system will show any discounts for any scholarships provided.
13. The system will automate the school's expenses, tax calculation, payroll service for the staff salaries and register any additional running costs.
14. The system will generate financial reports to help make a financial decision.
15. The system will **keep inventory records** for basic material like notebooks, markers, papers, pens and other stationery. It **will provide reminders when detecting any shortage** and thus ordering these materials in time. The cost will also be shown in the reports when ordering.

Management Features:

16. The teacher will be able to publish the grades of the students, it will provide a quick way to revise and update marks when necessary. The students and parents will be notified when a grade is published.
17. The system will allow principals to track and **monitor the teacher's log if any severe lag is detected** the principal will be notified to provide support and advice.
18. The system will **automate the task of building a term schedule for each class**. It will use an algorithm that will help meet the constraints including providing breaks for teachers and meeting the minimum classes required for each subject.

19. The system will provide reports for a section performance. These could help in assessing any learning issues.
20. The system will provide a compliant system for students, teachers and parents, and will allow to track any updates on the complaint status. The complaint can be forwarded to the intended responsible person.
21. The system will provide a **mechanism to easily take students' attendance** through a mobile feature by the class teacher. The class teacher will be given access to register attendance on the specified day. The parents will be urgently notified when their child is marked as unattended.
22. The system will integrate Biometric Attendance for the teachers and staff, it will show any delays and this would be reflected in their payments.
23. The school system will have a school public board for any announcements, emergency events and contests.
24. The system will provide a **"gamification" feature to engage students' participation**. Each student will have a score that will be grown automatically by achieving good grades, completing tasks and any endorsement from teachers. Highest achieving students of each section will be shown on the public school board on a monthly basis.

Registration Features:

25. The system will provide a mechanism for **registration online**. The parent can register a new child by submitting a new application. Parents will be able to directly upload necessary documents. The application can be approved by the principal and the registrar.
26. The system will be able to **indicate the number of seats that are available** for new students based on the schools capacity.
27. The registrar will have access to the students' and teachers' info and edit them. It will provide an automatic transfer application that includes necessary documents at once.
28. The system will have **an online application form for new teachers**. It will help in the hiring process, where relevant teachers can be filtered out and given points based on the school criteria, to help find suitable candidates.
29. The registration system will automate transcripts for students at the end of the semester. These will be sent to their parents directly and will be available in a print format.
30. The system will also provide **templates for extra certification** like recognition certificate, voluntary certificate, competition certificate, etc…

31. The system will provide services for parents like viewing transcripts and official documents.

Non-Functional Requirements:

- The system will focus on the usability aspect as it will be used by children and also staff with different backgrounds.
- The system will be maintainable, it would provide extensibility up to 10% increase in the school components for 5 years before an major update.
- The system will allow concurrent accesses with no interprion for 60% of the current users.
- The system will ensure security and data privacy.

Software development process:

In this project, we will use agile methodology due to its flexibility and repetitive nature, which makes it very suitable for dynamic environments such as school systems. Agile is the most ideal for these projects that require cooperation and continuity, which guarantees a system that meets the full needs required for this project.

Why Agile?

Flexibility and Iterative Development: The Agile model promotes repetition which enhances performance. This is important in that we will be able to provide some portions of the system for appraisal and assessment to verify that the system improves in congruence with the requirement of the school at any moment in time.

Customer Collaboration: Interaction with relevant parties (school management, teachers and students) will be regular so that we will be able to make some of the suggestions and be able to make the system more intuitive and practical. The continuous deployment allows for continuous feedback from the users and continuous testing.

Fast Delivery: Agile facilitates division of the project into sequences of smaller deliverables (sprints). This means that some portions of the software can be functional and deployed immediately, like simple basic student registration system and messaging features, addition of school schedules, even before the entire system is ready.

Release Plan:

We propose delivering the system into three iterative releases, each focused on progressively expanding the system's functionality:

- **Release 1**: **Basic Functions**- This phase focuses on the implementation of basic functions that solve the problems of the legacy system, the process of enrolling students and teachers, as well as messaging and helps to streamline the basic operations of the school immediately.
- **Release 2: Enhanced Features** – Additional smart features such as automatic class scheduling, grade publishing, payment automation, and detailed reporting for both

students and teachers. These enhancements will focus on improving efficiency and automating processes to reduce manual work.

- **Release 3**: **Final Polishing and Expansion** – This release will focus on refining the system, improving the user experience, addressing any remaining issues, and expanding the system with advanced features like a comprehensive analytics dashboard for school management and integration with external systems for advanced financial tracking or student performance analysis.

User & System Requirements:

1. **The EduSmart System shall allow students that are enrolled in a subject to post questions to a shared forum between the teacher and enrolled students.**
    1.1. At the start of the semester, the registrar creates subjects for each class and assigns the students to their subjects.
    1.2. The system should allow the teachers to create a forum for the subject and invite school students to join using their accounts.
    1.3. The system shall provide students with a "Raise a question" option in any forum that he/she is a part of.
        1.3.1. The system shall allow students who have not joined the forum to join.
    1.4. The question posted can be answered by any of the participants as a form of discussion.

2. **The EduSmart System shall allow parents to schedule meetings with their children's teachers.**
    2.1. The system shall allow the teacher to post their timetable and clearly state their available time slots for meetings. The available time slots are restricted between 8 AM and 6PM.
    2.2. The system shall allow parents to view the available time slots for meetings with their children's teachers after logging in with their parent account.
    2.3. The parent can submit a request for a meeting with the specified teacher, he/she can choose either online or face-to face meeting.
    2.4. The system shall notify the teacher about any requests.
    2.5. The parent shall be notified by email if the meeting is approved or rejected by the teacher.
    2.6. For online meetings, the system will then auto-generate a meeting link using a third party tool (Zoom).

3. **The EduSmart System shall allow teachers to build a balanced exam and mark them.**
    3.1. The system shall allow the teacher to create a new exam with a template, including number of questions, sequence, total mark, online/printed and time.

3.2. The system shall allow the teacher to add questions to the exam. The system allows teachers to search a question in the question bank and select from the previous questions.The system should allow the teacher to add a new question to the question bank and then add them to the exam. The teacher should then specify the difficulty level and expected solving time. The types of questions available are multiple choice questions, fill-in-blanks, essay questions, true or false questions and matching questions.

3.2.1. The system should allow the teacher to update an existing question.

3.2.2. The system should also allow the teacher to create a new version of an existing question.

3.2.3. The questions are stored in a centralized database within the system's infrastructure.

3.3. The system shall allow teachers to distribute marking weight to the exam questions.

3.4. The system should allow teachers to determine the correct answers for the objective questions such as multiple choice, fill-in-blanks, true or false, etc.

3.5. The system shall be able to automark the objective questions in the exam.

3.6. The system should allow teachers to provide feedback for essay questions and manually grade them.

3.7. The system should allow the teacher to generate the exam either as an online or a printed format. For a printed format the teacher has the option to choose from 3 templates.

3.8. The system should verify whether the exam is balanced or not according to the pedological metrics adapted and implemented by the school system, including time constraints, distribution of questions on different levels of difficulties, diversability, and detailed description of these metrics are documented on the teacher pedological guideline .

3.8.1. If the exam does not meet the metrics the system shall notify the teacher with the problem.

3.9. The system should create the exam in a period of 2 seconds after getting the questions inserted.

4. The EduSmart System shall build a weekly plan for each class, provided information from the class's teachers.

4.1. The system shall request information from every subject teacher in the middle of the week.

4.1.1. The information provided should contain a description for the weekly materials and tasks for the subject.

4.2. The system shall send the finished plan to the principal to approve or disapprove it and write comments on what needs to be changed

4.3. The system should notify every teacher who's part has been commented on by the principal.

4.3.1. The system should allow the teacher to view principal comments in case of the unapproval to the plan.

4.3.2. The system should allow the teacher to change the materials provided according to the principle comments.

4.4. The system shall be able to create the next weekly plan by Friday.

4.5. The system shall allow students and their parents to view the weekly plan and notify them when it’s available to be viewed.

5. The EduSmart System shall allow parents to online register their children in the school.

5.1. The system shall allow parents to renew the registration of a current student for a new academic year.

5.2. The system shall allow parents to apply for a new registration for their children.

5.3. The system shall enable parents to upload the necessary documents for the registration, such as :

- The legal guardian’s ID photo.
- The child birth certificate photo.
- A photo of the child.
- Previous certificates for transfer students and a transfer letter from its current school.
- A paper that contains more information about the child and the family history (it contains the child age, name , the former school name, the place of birth, the date of birth, the nationality, the home address, the home phone, the history of chronic diseases in the family ,how many siblings does the child have , the arrangement of the child among the siblings.. etc.) that is needed by school.

5.4. The system should enable the school principal to determine the capacity of each grade.

5.5. The system shall provide the principal and the registrar access to view and approve the registration application.

5.5.1. The system should check the availability of seats in the grades, and display for the principal and the registrar the current number of registered students and the current availability.

5.5.2. Upon Approval the system shall update the grades' seats.

5.6. The system shall notify the parent with the school administration decision about the acceptance or rejection.

6. The EduSmart System shall provide a mobile feature to record students' attendance.

6.1. The system shall allow the class teacher to take student attendance through an integration with a mobile feature. The headteacher will be responsible to take attendance of the class by viewing a list of students and choosing unattended.

6.2. The system shall give the class teacher access to register attendance on the specified day.

6.3. The system shall notify the parents when the class teacher is done taking attendance within 5 minutes if their child has been marked as unattended.

7. The EduSmart System shall generate transcripts and certifications for each student.

7.1. The system should automatically generate transcripts for students at the end of each semester.

7.1.1. Transcripts should include student grades, attendance records, and overall performance.

7.1.2. The system should send digital copies of transcripts to parents and make printable versions available.

7.2. The system shall provide templates for additional certifications.

7.2.1. Templates should include recognition certificates for academic excellence, participation, and other achievements.

7.3. The system shall allow teachers and administrators to generate custom certifications using pre-designed templates.

8. The EduSmart System should engage students' participation by providing rewards points for each student.

8.1. The system shall provide a “gamification” feature to enhance student engagement.

8.1.1. Each student has a score that grows automatically based on good grades, completed tasks, and teacher endorsements.

8.2. The system shall calculate points for all students.

8.3. The system shall display the points for all the students each month.

8.3.1. Monthly top-achieving students in each section will be highlighted on the school’s public board.

8.3.2. Parents receive notifications if their child ranks among the top achievers.

8.4. The system shall allow teachers to assign additional rewards or penalties to students' scores.

8.5. The system shall maintain a leaderboard that updates monthly to encourage healthy competition among students.

9. The EduSmart System shall provide an online application for new teachers to apply.

9.1. The system shall allow the school principal to post for a new position. It should describe minimum qualifications, description and a deadline.

9.2. The system shall allow prospective applicants to apply for an open position.

9.2.1. Applicants fill out personal details such as average, graduation year, years of experience, certifications, qualifications, and experience information.(for application)

9.2.2. The system allows applicants to upload required documents including resumes and certifications.

9.3. The system shall filter applications by graduation year, years of experience and certifications.

9.4. The system shall provide the principal with a dashboard to review filtered applications.

9.5. The system shall allow the principal to mark applications as shortlisted, rejected, or accepted.

9.6. Notifications shall be sent to applicants about their application status.

9.7. The system shall allow the principal to schedule interviews for shortlisted applicants.

9.7.1. Applicants choose between online or face-to-face interview options.

10. The EduSmart System shall automate the building of a term schedule for each class.

10.1. The system shall allow administrators to define scheduling constraints.

10.1.1. Constraints include teacher availability, subject-specific requirements, and break times.

10.2. The system shall generate a balanced term schedule using predefined algorithms.

10.2.1. The algorithm ensures no conflicts in teacher or room assignments.

10.3. The system shall allow administrators to review and make manual adjustments to the generated schedule.

10.4. The system shall notify teachers, students, and parents about the finalized term schedule.

10.5. The system shall dynamically update the schedule in response to changes, such as teacher unavailability or school events.

10.5.1. All affected stakeholders receive notifications of any updates.

Cost and Effort Estimation:

Effort estimation

To determine the effort and cost estimation perfectly, we divided the project into key components, to provide a comprehensive estimation for each one.

Effort estimation: The Work Breakdown Structure (WBS):

- Requirement Analysis: review and documentation.
- Design: database design, UI/UX design, and architectural design.
- Development: Coding, and module development.
- Testing: System testing, and user acceptance testing.
- Deployment and Maintenance: Deployment on the server, and user training.

Effort estimation for each phase (person-Days):

Phase	***Days – person days***	***Justification***
Requirement Analysis	20 days	In this phase, we'll be gathering detailed requirements form stakeholders
Design	30 (month)	The designing phase will contain creating the workflows, wireframes, and the whole system's architecture
Development	90 (3 months)	The most important and the longest phase, it involves building the modules (registration, payments, schedules, etc.)

Testing	30 (month)	The testing phase is for ensuring that all the functionalities work as expected, and for fixing any bugs could occur.
Deployment	10	For deploying and configuration on servers.
Maintenance	30 (month)	The post-deployment support phase, for bug fixes, and other upgrades.
Total	**210 days**	

For a team of 4 developers working full-time (let's say 20 days per month), the total project duration is approximately *3 months*

Total duration = (210 (Person-Days))/(4 (developers)× 20 (work days per month for a developer))= 3 months

Cost Estimation

To ensure a comprehensive cost estimation, the effort estimation provided was aligned with the hourly rates of the team roles. The total cost was calculated based on the effort (person-days) required for each phase.

1. Labor Cost

Team Roles and Hourly/Daily Rates

The following roles were considered for the project, along with their respective rates:

Team Roles and Hourly/Daily Rates:

Role	Hourly Rate ($)	Daily Rate ($)	Number of Employees	Justification	Effort (Person-Days)	Total Cost for each employee $
Project Manager	50	400	1	to guide the team and focus on project phases	10% (21 days)	$8,400
Backend Developers	35	280	2	Responsible for core system logic and integration	40% (84 days) 42 each	$23,520
Frontend Developers	30	240	2	to focus on UI/UX design and frontend functionality	30% (63 days) 31.5 each	$15,120

Quality Assurance Engineer	25	200	1	Handles system testing and post-deployment support	20% (42 days)	$8,400

Total Labor Cost: 8400 + 23520 + 15120 + 8400 = **$55,440**

2. Supplementary Costs

This section will provide additional infrastructure, training, and contingency costs:

Cost Breakdown Components	**Amount $**	**Justification**
Hosting & Infrastructure	5000	cloud servers , software licenses, hardware
Miscellaneous	2000	preparing initial documentation and training users on the system
Contingency	6000	Buffer for (unexpected delays , unforeseen expenses , additional features)

Total Supplementary Costs : 5000 + 2000 + 6000 = **$13,000**

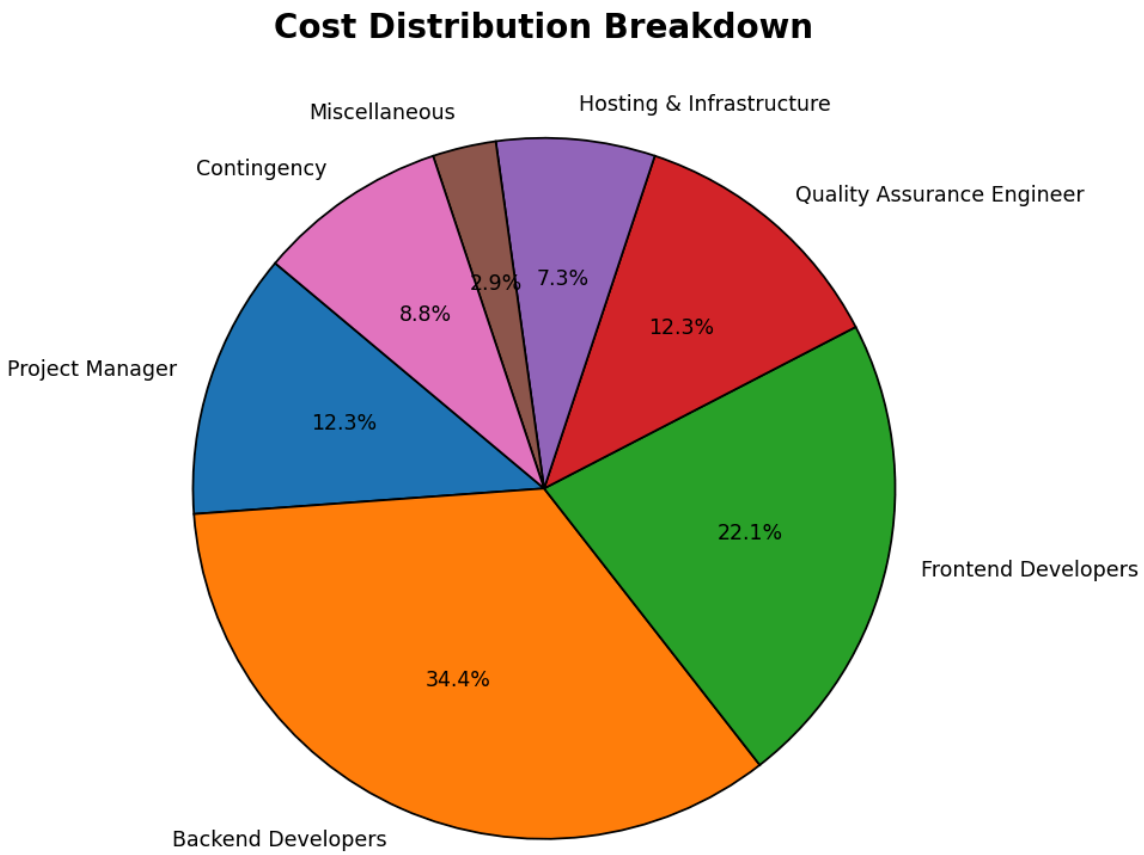
Total Project Cost

The total cost of the project, including labor and additional costs, is calculated as:

Labor Cost: $55,440

Additional Costs: $13,000

Total Project Cost: $68,440



Scenario Analysis:

1. Create New Exam Scenario (Sarah Hassouneh)

This template is taken from *SOFTWARE ENGINEERING, Ninth Edition, Ian Sommerville* book, chapter 4, section 4.5.3.

Initial Assumption:

The teacher is logged in into their teacher's account, the subject is already created with its question bank, and the exam metrics are set by the principal.

Normal:

The teacher selects the subject he/she wants to create an exam for and then chooses the "Create an Exam" option. The teacher sets up the exam settings including number of questions, total marks and assigned time. The system validates these settings and then the teacher can start adding questions.

The teacher adds a question one by one from the question bank. The teacher assigns weight for each added question and saves a draft after each addition. After adding all questions the teacher chooses a preview option. In the preview window he/she can re-orders the question sequence or modify a weight for any question. The teacher then submits the exam where it would be validated based on exam types, difficulty, total marks, and assigned time.
The teacher selects the "Export as PDF" option and chooses a template, then the exam is downloaded as pdf. The teacher can also select an online exam and schedule a time for the exam.

What Can Go Wrong:

1. Invalid Settings: The teacher enters less than 2 questions or sets the assigned time below 5 minutes. The system highlights the error and prompts for correction.

2. Questions are not available or empty test banks: The question bank may not have intended questions and so the teacher can create new questions or update existing ones.

3. Metrics are not met: The exam does not meet metrics The system notifies the teacher, with the metrics that are not met.

4. Connection Lost during Exam Creation: The system saves the last draft locally. When the connection is restored, the teacher can continue from the last draft

Other Activities:

- The teacher can modify the exam or the settings before final submission and re-validate.
- The system saves any added question into the question bank, that can be accessed by any teacher teaching the subject.

System State on Completion:

The exam is saved as "Complete" under the selected subject.

2. Register a Child Scenario (Dana Hafitha)

Initial Assumption:

The parent is already on the main page of the system and the “register a child” option is available.

Normal:

After the parent goes to the site in order to register a child, the parent chooses “register a child” option from the main page, the system then open another page that first requires full name and the id of the parent and from it the system searches the database on the id to make sure the email is not there , if the email is not in the database the system ask for the email of the parent in order to contact with the parent, then send a verification link through email and asks the parent to verify the email, after confirming the email successfully, the system then proceeds to the next page where it requires giving the personal information of the child that includes the full name, the date of birth, gender, and uploading the necessary documents including the birth certificate, parent id, a multiple photos of the child, a previous certificates for the child and a transfer letter from its current school and choosing the grade of the child, after that it shows message that says “the registration has been done successfully, Please continue checking emails for any updates”.

What Can Go Wrong:

Alternative scenarios:-

1. Grades are full : The parent chooses the grade of the child, the system finds that the grade chosen is full, then the system displays a message saying “Sorry the grade you have chosen is full” and closes the registration page.

2. The email inserted is already in the database: The parent don’t need to continue filling if the email is already in the database and it’s autofilled by the system.

Error scenarios:-

1. The email inserted is not valid: The parent email is not a valid email, the system asks the parent to re enter his email.

2. Lost connection during the process: The system saves the last actions of the parent, displays a message saying “Connection is lost, please reconnect and try again” and when the connection is restored, the parent continues filling the form without anything lost.

Other Activities:

There are no other activities that the parent could do.

System State on Completion:

The system saves the form in the database with the status of it whether it's pending or accepted or rejected.

3. Assign and Mark Assignments (Doaa Hatu)

Initial Assumption:

- The instructor is logged into their account.
- The course is already created.
- The assignment details are prepared.

Normal Flow:

1. The instructor selects the course and clicks "Create Assignment."
2. The instructor enters assignment details (title, description, deadline).
3. The system validates the inputs and successfully saves the assignment.
4. The system notifies students about the new assignment.
5. Students view and submit their assignments by the deadline.
6. The instructor views submissions, grades them, and updates the grades in the system.
7. Notifications are sent to students about their grades.

Alternative Flows:

1. **Assignment Modification Before Deadline:**
    - If the instructor notices errors in the assignment details, they can modify the assignment before the deadline.
    - The system notifies students about the updates.
2. **Resubmissions by Students:**
    - If a student identifies an error in their submission before the deadline, the system allows them to resubmit the assignment.
    - The new submission overwrites the previous one.

Error Flows:

1. **Incomplete Details:**
    - The instructor enters incomplete details (e.g., missing deadline).
    - The system displays error messages and prompts the instructor to correct the inputs.
2. **Missing Submissions:**
    - A student fails to submit the assignment before the deadline.
    - The system marks the submission as "Late" or "Missing" and notifies the instructor.
3. **Grading Interruption:**
    - The grading process is interrupted (e.g., system crash or logout).
    - The system auto-saves grading progress and allows the instructor to resume later.
4. **Notification Failure:**
    - Notifications to students fail due to system errors.
    - The system retries sending notifications and logs the failure for administrative review.

Other Activities:

- The system provides a summary report of submissions and grades.
- The instructor can view analytics on student performance for the assignment.

System State on Completion:

- The assignment is marked as "Complete."
- All grades are updated in student records.
- Notifications about grades are successfully sent to students.

4. Submit an Application for a Teaching Position(Diana Naseer)

Initial Assumption:

The applicant has access to the internet and is on the "Careers" page of the school management system, ready with the necessary documents in DOCX format to upload.

Normal Flow:

The applicant navigates to the "Careers" section of the school management system where they find a list of available teaching positions. Each position includes a detailed description, required qualifications, and application deadlines. The applicant selects a position and is prompted to log in or create an account. For new users, account creation involves entering details such as name, email, phone number, and creating a password. Upon logging in, the applicant fills out an application form that includes fields for personal information, academic qualifications, and professional experience. The system then prompts the applicant to upload supporting documents such as a resume, certificates, and letters of recommendation. All files must meet specific format and size requirements. After completing the form and uploading documents, the applicant submits their application and receives a confirmation with a unique reference number.

What Can Go Wrong

Error Flow

1. **Unsupported File Format or Size**: If the applicant attempts to upload a document in an unsupported format, such as a JPG, or if the file size exceeds the system's limit, the system will display an error message: "Unsupported file format or file size too large. Please upload a PDF or DOCX file under 10 MB." The application cannot proceed until the file meets the specified criteria.
2. **Missing Mandatory Fields**: If the applicant skips one or more required fields in the application form, the system will prevent form submission and highlight the missing fields with an error message: "This field is required."

Alternative Flow

3. Internet Connection Loss: In case of an internet disconnection, the system will automatically save the application's current progress. This feature allows the applicant to resume the application from where they left off once the internet connection is restored.
4. Incomplete Applications: If the applicant submits an application with errors or missing details, the system will save it in "Draft" status and notify the applicant to complete the necessary corrections. The applicant can then access the draft, make the required edits, and resubmit before the deadline.

Other Activities:

The applicant can save their progress at any time and return later to complete the application. They can also update their profile details, edit the application, or replace uploaded documents before submission. The system provides the option to withdraw a submitted application. Status updates are provided through the dashboard, where the application can appear as "Under Review" or "Incomplete," with notifications for important deadlines and status changes.

System State on Completion:

Once submitted, the application is marked as "Submitted" and appears as "Under Review" on the applicant's dashboard. A confirmation email with a unique reference number is sent, and all related notifications are successfully delivered.

5. Update students attendance (Leen Daraghmeh)

Initial Assumption:

The teacher is logged into their account using valid credentials, and the system ensures that the attendance module is operational.

Normal Flow:

The teacher navigates to the attendance records module. After accessing the module, the teacher selects the class and date for marking attendance. The teacher marks each student as present or absent using the interface provided. Once all entries are completed, the system saves the attendance records securely. Students can view their updated attendance records through their accounts, and notifications are automatically sent to the parents of absent students to keep them informed.

What Can Go Wrong

Alternative Flows:

1. **Internet Connection Issue:**
    - **Description:** The teacher attempts to log attendance but encounters an internet connection issue.
    - **Resolution:** The system detects the issue and switches to offline mode, allowing the teacher to log attendance locally on the device. The system temporarily stores the attendance data. Once the connection is restored, the system automatically syncs the offline data with the central database.
    - **Outcome:** Notifications are then sent to parents of absent students, ensuring no delays in communication.
2. **Duplicate Attendance Records:**
    - **Description:** The teacher attempts to mark attendance for a class that already has records for the selected date.
    - **Resolution:** The system warns the teacher about the duplication and offers options to overwrite or merge the records.
    - **Outcome:** The teacher reviews the existing records and chooses the appropriate action.

Error Flow:

1. **System Timeout:**
    - **Description:** The system logs the teacher out due to inactivity or session timeout.
    - **Resolution:** The teacher is redirected to the login page. After logging back in, the teacher is prompted to resume the attendance-taking process.
    - **Outcome:** The system retrieves and restores any previously entered attendance data to ensure no progress is lost.

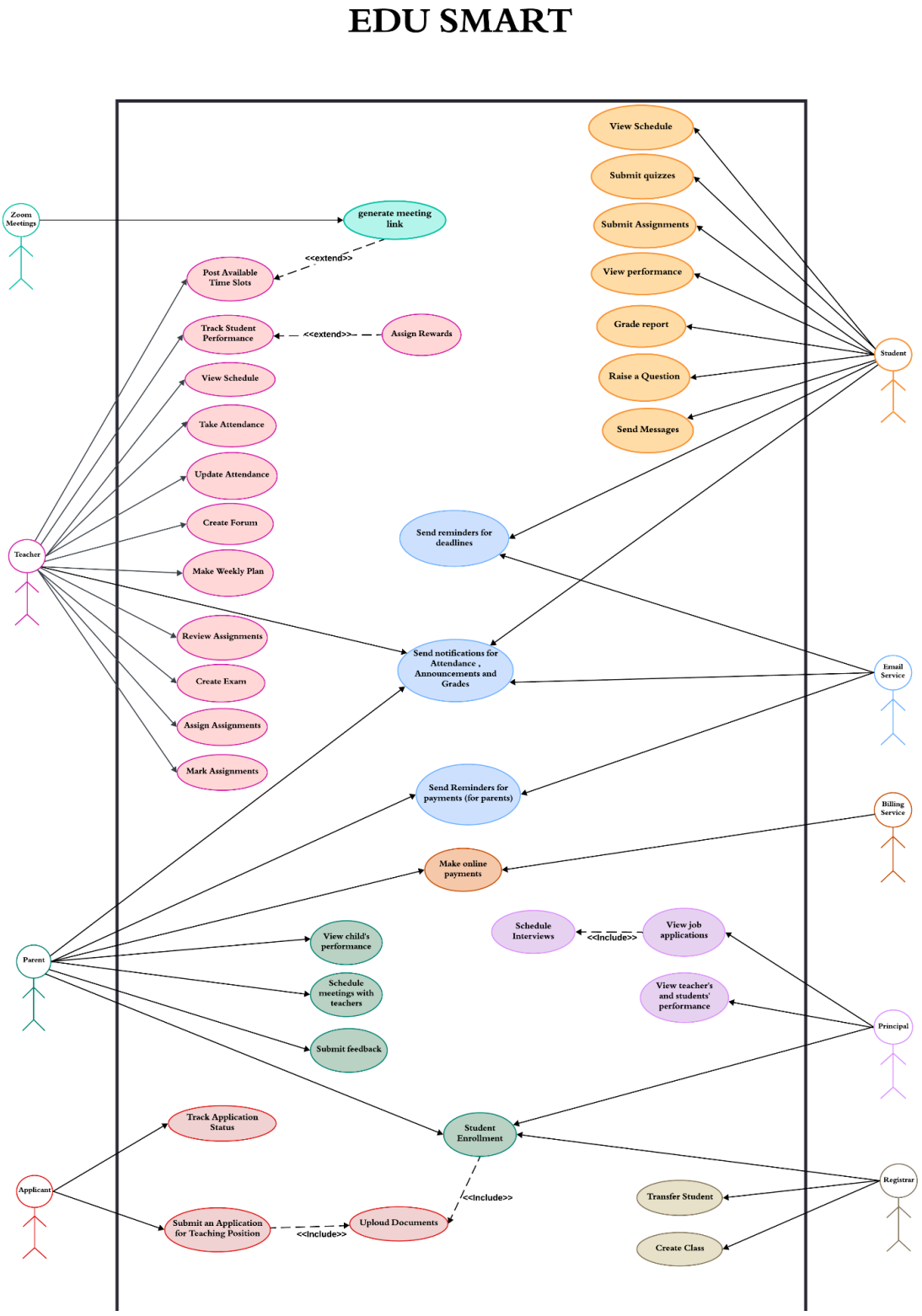
Other Activities:

- The teacher can review and edit attendance records after saving, provided they have the necessary permissions.
- The system generates attendance reports that can be accessed by the teacher or principal.
- Notifications are sent to parents regarding absences.

System State on Completion:

- Attendance records are marked as "Complete" for the selected class and date.
- The system ensures data consistency and syncs offline entries with the central database when necessary.

Actors and Use case (Diagram)[1]



---

[1] View on LUCID chart

Use case Description

1. Use-Case: Create New Exam (Sarah Hassouneh)

1.1. Brief Description

This use case allows the teacher to create a new exam for the students. The exam is built with multiple questions.

1.2. Actor

The primary actor for this use case is the teacher.

1.3. Stakeholders

- Principal and administration: The principal and administration are responsible for setting metrics for an exam.
- Students: The exam created will be solved by students.

1.4. Preconditions

- Teacher is logged into his/her account.
- Metrics for a balanced exam are set by the principal.
- The subject is already created in the system to enable the creation of exams for it. Each subject has a question bank.

1.5. Flow of Events

1.5.1. Basic Flow

1. The teacher starts by accessing the specific subject he/she wants to create an exam for, under this subject the teacher accesses the “Create an Exam” option to start creating the exam.
2. The teacher sets settings for the exam including number of questions, total mark and assigned time for the exam. The system will ensure that the number

of questions is at least 2 (based on school's metrics) and assigned time is at least 5 minutes.

3. The stem validates the sayings and if all is valid the teacher now can start by adding questions to the exam. The teacher can search from the question bank and select a question from existing questions. Each question is marked with a difficulty level (easy, medium, hard), expected solving time and type (multiple choice question, fill-in-blanks, essay question, true or false question or matching question). After choosing the question, the teacher assigns weight to it. A new draft is saved after each question.
4. The teacher can keep adding questions until the number of questions is reached.When the number of questions set in the settings is reached, no further questions can be added.
5. Finally, the teacher previews the exam, he/she can re-orders the question sequence or modify a weight for any question and then submit the exam for validation.
6. The system will validate exam based on the following, in compliance with the school's metrics:
   - The exam shall have at least 2 questions.
   - The exam questions shall be of different types and different difficulty levels.
   - The exam expected time - calculated by the sum of expected times of all added questions - shall not exceed the assigned time by more than 5%.
   - The total weights of all exam questions shall match the total mark as set in the exam settings.

   The detailed description of the exam metrics is documented on the teacher pedological guideline.
7. The system will save the exam as a "Complete" exam under the subject module.
8. The teacher can export a "Complete" exam as a printed format with 3 templates to choose from, the exam will then be downloaded in a pdf format, or he/she can choose to post it as an online exam and specify date and time of exam.

1.5.2. Alternative Flows

Note : This writing style of use case description follows the style of (Applying UML and patterns by Craig Larman) where (2a) represents an alternative flow of the second point in the main alternative, and no need to rewrite all flow.

2a. Invalid settings: If the number of questions is less than 2 or assigned time is less than 10, the system will highlight the error to be edited before adding questions.
3a. The teacher can choose to add a new question to the question bank. The teacher adds the question, specifies expected solving time, difficulty level, type (multiple choice question, fill-in-blanks, essay question, true or false question or matching question). For objective questions the teacher shall specify the correct answer.
3b. The teacher can choose to update an existing question in the question bank and save it as a new version.
3c. The teacher wants to add a new question after reaching the limit. The system shall notify the teacher and give two options: edit the settings or choose a question to replace.
5a. The exam does not meet metrics, and thus the system shall notify the teacher with the metrics that are not met. The teacher can then edit the questions or settings to meet metrics.

1.6. Postconditions (Success Guarantee)

- A new exam is created and saved under the subject.

1.7. Special Requirements

- The system should save drafts of the exam in a local cache.
- The system should validate the exam in a period of 1 seconds after getting the questions inserted.
- The system should export the exam in a printed format in a period of 2 sec.

1.8. Extension Points

- Custom Exam Templates: Teachers can design custom print templates in addition to the three predefined templates.
- Question Randomization: The teacher can enable a setting to randomize the sequence of questions in the online exam for each student.

2. Use-Case: Register a Child (Dana Hafitha)

2.1. Brief Description:

This use case allows the user to register a child in the school system.

2.2. Actor:

The actor is the parent of the child

2.3. Stakeholders:

- the principle to view the applications and approve them
- the registrar to manage the applications.

2.4. Preconditions

There are no pre-conditions for that use case.

then send a verification link through email and asks the parent to verify the email, after confirming the email successfully, the system then proceeds to the next page where it requires giving the personal information of the child that includes the full name, the date of birth, gender, and uploading the necessary documents including the birth certificate, parent id, a multiple photos of the child, a previous certificates for the child and a transfer letter from its current school and choosing the grade of the child, after that it shows message that says “the registration has been done successfully, Please continue checking emails for any updates”.

2.5. Flow of Events

2.5.a. Basic Flow – register a child

2.5.a.1. The parent selects to “register a child”

2.5.a.2. The system shows a page asking for the ID and the full name of the parent.

2.5.a.3. The parent fills the information asked.

2.5.a.4. The system searches the database for the name and the ID, since the system will not find the Id the system will ask for the email of the parent.

2.5.a.5. the parent inserts the email.

2.5.a.6. The system sends a verification email into the arent email.

2.5.a.7. The parent verifies the email.

2.5.a.8. The system displays a form asking for the child's full name, birth certificate, parent id, multiple photos of the child, a previous certificate for the child and a transfer letter from its current school, and asks to select the grade of the child.

2.5.a.9. The parent insert all the required information and upload all the documents asked then selects “register”

2.5.a.10. The system saves all the information inserted in the form and displays a message saying “the registration is completed please continue checking the email for any updates”.

2.5.b. Alternative Flows

2.5.b.1. The parent email is already in the database.

- The parent selects to “register a child”
- The system shows a page asking for the ID and the full name of the parent.
- The parent fills the information asked.
- The system searches the database for the name and the ID, and finds it in the database and auto inserts the email.
- the parent continue to the main form
- The system displays a form asking for the child's full name, birth certificate, parent id, multiple photos of the child, a previous certificate for the child and a transfer letter from its current school, and asks to select the grade of the child.
- The parent insert all the required information and upload all the documents asked then selects “register”
- The system saves all the information inserted in the form and displays a message saying “the registration is completed please continue checking the email for any updates”.

2.5.b.2. The grade chosen for the child is full

- The parent selects to “register a child”
- The system shows a page asking for the ID and the full name of the parent.

- The parent fills the information asked.
- The system searches the database for the name and the ID, since the system will not find the Id the system will ask for the email of the parent.
- The parent inserts the email.
- The system sends a verification email into the arent email.
- The parent verifies the email.
- The system displays a form asking for the child's full name, birth certificate, parent id, multiple photos of the child, a previous certificate for the child and a transfer letter from its current school, and asks to select the grade of the child.
- The parent insert all the required information and upload all the documents asked then selects “register”
- The system finds that the grade chosen by the parents is full and displays the message “Sorry the grade you choose is full” to the parent.

2.5.b.3. The email inserted is not valid

- The parent selects to “register a child”
- The system shows a page asking for the ID and the full name of the parent.
- The parent fills the information asked.
- The system searches the database for the name and the ID, since the system will not find the Id the system will ask for the email of the parent.
- The parent inserts the email.
- The system checks if the email is valid and asks for the email since the email is not valid by displaying the following message: “the email you inserted is not valid, please enter a valid email address”.
- the parent re-insert the email.
- The system sends a verification email into the parent email.
- The parent verifies the email.
- The system displays a form asking for the child's full name, birth certificate, parent id, multiple photos of the child, a previous certificate for the child and a transfer letter from its current school, and asks to select the grade of the child.

- The parent insert all the required information and upload all the documents asked then selects “register”
- The system saves all the information inserted in the form and displays a message saying “the registration is completed please continue checking the email for any updates”.

2.5.b.4. The connection is lost

- The parent selects to “register a child”
- The system shows a page asking for the ID and the full name of the parent.
- The parent fills the information asked.
- The system searches the database for the name and the ID, since the system will not find the Id the system will ask for the email of the parent.
- The parent inserts the email.
- The system sends a verification email into the arent email.
- The parent verifies the email.
- The connection is lost and the system saves the version in the database.
- The connection returns and the parent reloads the page.
- The system gets the before version from the database and continue the modification on it then displays a form asking for the child's full name, birth certificate, parent id, multiple photos of the child, a previous certificate for the child and a transfer letter from its current school, and asks to select the grade of the child.
- The parent insert all the required information and upload all the documents asked then selects “register”
- The system saves all the information inserted in the form and displays a message saying “the registration is completed please continue checking the email for any updates”.

2.6. Post-conditions

The system will update the capacity of the grade chosen in the register process and send an approval email to the parent.

2.7. Special Requirements

2.7.1. The system has to search the database for the email in 1-2 seconds.

2.7.2. The system has to send the verification email within 2-3 seconds.

2.8. Extension Points

There are no extinction points in that use case.

3. Use-Case: Assign and Mark Assignments (Doaa Hatu)

3.1 Use Case: Assign Assignments

3.1.1. Brief Description

This use case describes the process through which an instructor assigns assignments to students. The system validates and saves the assignment details, notifies students, and provides them with access to the assignment.

3.1.2. Actors

- **Instructor**: Responsible for creating and managing assignments.
- **Student**: Responsible for viewing and submitting assignments.

3.1.3. Preconditions (Entry Condition)

1. The instructor is logged into the school management system.
2. The assignment details (title, deadline, description) are prepared.
3. Students are enrolled in the relevant course.

3.1.4. Basic Flow

Assign Assignment

1. The instructor selects the course.
2. The instructor clicks "Create Assignment."
3. The instructor enters assignment details, such as title, description, and deadline.
4. The system validates the entered details.
5. The system saves the assignment and notifies students.

3.1.5. Alternative Flows

1. **Incomplete Assignment Details:**
    - If the instructor enters incomplete or invalid assignment details, the system highlights the missing or incorrect fields and prompts the instructor to correct them.
2. **Notification Delivery Issues:**
    - If the system encounters a failure in sending notifications to students, it logs the error and retries delivery until successful.

3.1.6. Post-conditions (Exit Condition)

1. The assignment is successfully created and accessible to students.
2. Notifications are sent to all students regarding the new assignment.

3.2. Use Case: Mark Assignments

3.2.1. Brief Description

This use case describes the process through which an instructor views and grades students' submissions for an assignment, and the system updates and notifies students about their grades.

3.2.2. Actors

- **Instructor**: Responsible for grading the assignments.
- **Student**: Receives grades for submitted assignments.

3.2.3. Preconditions (Entry Condition)

1. The assignment has been created and assigned to students.
2. Students have submitted their assignments.

3.2.4. Basic Flow

1. The instructor logs into the school management system and selects the relevant course.
2. The instructor views the list of submitted assignments.
3. The instructor grades the assignments one by one.
4. The system updates the grade records.
5. The system notifies students about their grades.

3.2.5. Alternative Flows

1. **Grading Process Interrupted:**

- If the grading process is interrupted (e.g., system crash or logout), the system automatically saves progress and allows the instructor to resume grading from the last completed point.

2. **Late or Missing Submissions:**

   - If a student fails to submit their assignment before the deadline, the system flags the submission as "Late" or "Missing" and notifies the instructor.

3.2.6. Post-conditions (Exit Condition)

1. Grades are successfully recorded and visible to students.
2. Notifications about the grades are sent to all students.

3.2.7. Special Requirements

1. The system must auto-save progress during grading to prevent data loss.
2. The system must log notification delivery statuses for troubleshooting purposes.

3.2.8. Extension Points

1. Integration with an external LMS for advanced analytics and reporting.

4. Use-Case: Submit an Application for a Teaching Position (Diana Naseer)

4.1 Brief Description

This use case describes the process through which an applicant submits an application for a teaching position via the school's career page. This includes uploading necessary documents and completing required forms.

4.2 Actor

Primary Actor: Applicant.

4.3 Stakeholders

School HR and Administration: Responsible for reviewing and processing the applications.

Applicants: Individuals applying for the teaching positions.

4.4 Preconditions

4.4.1 The required documents (resume, certificates, etc.) are prepared and saved in the correct format.

4.5 Flow of Events

4.5.1 Basic Flow

1. **Accessing the Careers Section**: The applicant enters the "Careers" section of the school management system.

2. **Selecting a Position**: They browse the list of available teaching positions, reviewing details like qualifications needed and responsibilities for each position, then select the one they wish to apply for.

3. **Account Interaction**: The applicant clicks "Apply Now." If not already logged in, they are prompted to log in or register. Registration requires entering personal information

like name, email, and creating a password.

4. **Filling Out the Application Form**: Upon logging in or registering, the applicant completes the application form, which includes sections for personal details, academic qualifications, and professional experience.

5. **Uploading Documents**: The applicant uploads required documents such as resumes, certificates, transcripts, and letters of recommendation. The system will specify acceptable formats and size limits.

6. **Document Verification**: The system automatically checks the uploaded documents for compliance with format and size requirements and confirms the successful upload.

7. **Review and Submission**: The applicant reviews all entered information and uploaded documents, making any necessary adjustments before submitting the application.

8. **Confirmation and Status Update**: Upon submission, the system sends a confirmation email to the applicant with a unique reference number and updates the application status on the dashboard to "Under Review."

4.5.2 Alternative Flows

4.5.2.1 Returning Applicants: If the applicant has previously applied, they are automatically directed to a pre-filled form upon logging in. This form contains previously submitted personal and professional details, which the applicant can review and update as necessary to ensure all information is current and accurate.

4.5.2.2 Document Compliance Errors: Should the applicant upload a document that fails to meet the required format or size specifications, the system will immediately display a notification specifying the error (e.g., "This file exceeds the maximum allowed size" or "Invalid file format"). The applicant is prompted to upload a new file that adheres to the stated requirements before they can proceed with the application.

4.5.2.3 Form Completion Errors: If the application form is incomplete or contains errors (such as missing mandatory fields), the system will prevent the application from being submitted. It will highlight the incomplete or incorrect fields, providing specific instructions or reasons (e.g., "This field is required" or "Invalid entry"). The applicant must resolve these issues to enable the submit button.

4.6 Postconditions

4.6.1 The application is successfully submitted and marked as "Under Review" on the applicant's dashboard.

4.6.2 The applicant receives a confirmation email.

4.7 Extension Points

4.7.1 Document Management Options**:** Applicants can manage and replace documents in subsequent applications.

4.7.2 Application Status Updates**:** Applicants receive real-time updates about the status of their applications through notifications.

5. Use-Case: Update Student Attendance (Leen Daraghmeh)

1. Brief Description

This use case allows teachers to record and update student attendance. It ensures attendance records are saved securely, and notifications are sent to parents about student absences.

1.1. Actor

1.1.1 Teacher

1.1.2 Parent

2. Preconditions (Entry Condition)

2.1 The teacher must be logged into the system.

2.2 The attendance module must be accessible and functioning correctly.

2.3 Class and student data are preloaded into the system.

3. Flow of Events

3.1 Basic Flow - Normal Attendance Marking

3.1.1 From the dashboard menu the teacher navigates to the attendance records module

3.1.2 The system displays a list of available classes, and the teacher selects the desired class and date for marking attendance.

3.1.3 The system loads the list of students in the selected class, allowing the teacher to mark each student as present or absent.

3.1.4 The teacher completes marking attendance and reviews the entries to ensure accuracy.

3.1.5 The teacher submits the attendance data, and the system securely saves the records in the database.

3.1.6 Students are immediately granted access to view their updated attendance records via the student portal.

3.1.7: The system automatically sends notifications to the parents of absent students. These notifications include detailed information such as the class name, date of absence, and the attendance status of the student.

3.1.8 The use case ends.

4. Alternative Flows

4.1 Internet Connection Issue

4.1.1 The teacher attempts to log attendance but encounters an internet connection issue.

4.1.2 The system detects the issue and switches to offline mode.

4.1.3 The teacher logs attendance locally on the device.

4.1.4 The system temporarily stores the attendance data.

4.1.5 Once the connection is restored, the system syncs the offline data with the central database.

4.1.6 Notifications are then sent to parents of absent students, ensuring no delays in communication.

4.2 Duplicate Attendance Records

4.2.1 The teacher attempts to mark attendance for a class that already has records for the selected date.

4.2.2 The system warns the teacher about the duplication and provides options to overwrite or merge the records.

4.2.3 The teacher reviews the existing records.

4.2.4 The teacher chooses the appropriate action (overwrite or merge).

5. Post-conditions (Exit Condition)

5.1 Attendance records are securely saved in the system for the selected class and date.

5.2 Notifications about absences are successfully sent to the parents of absent students.

5.3 Students can view their updated attendance records through their accounts.

6. Special Requirements

6.1 The system must securely store all attendance data to ensure confidentiality and integrity.

6.2 Notifications must be sent to parents within five minutes of saving the attendance data.

6.3 The system must support offline mode to ensure uninterrupted attendance logging during connectivity issues.

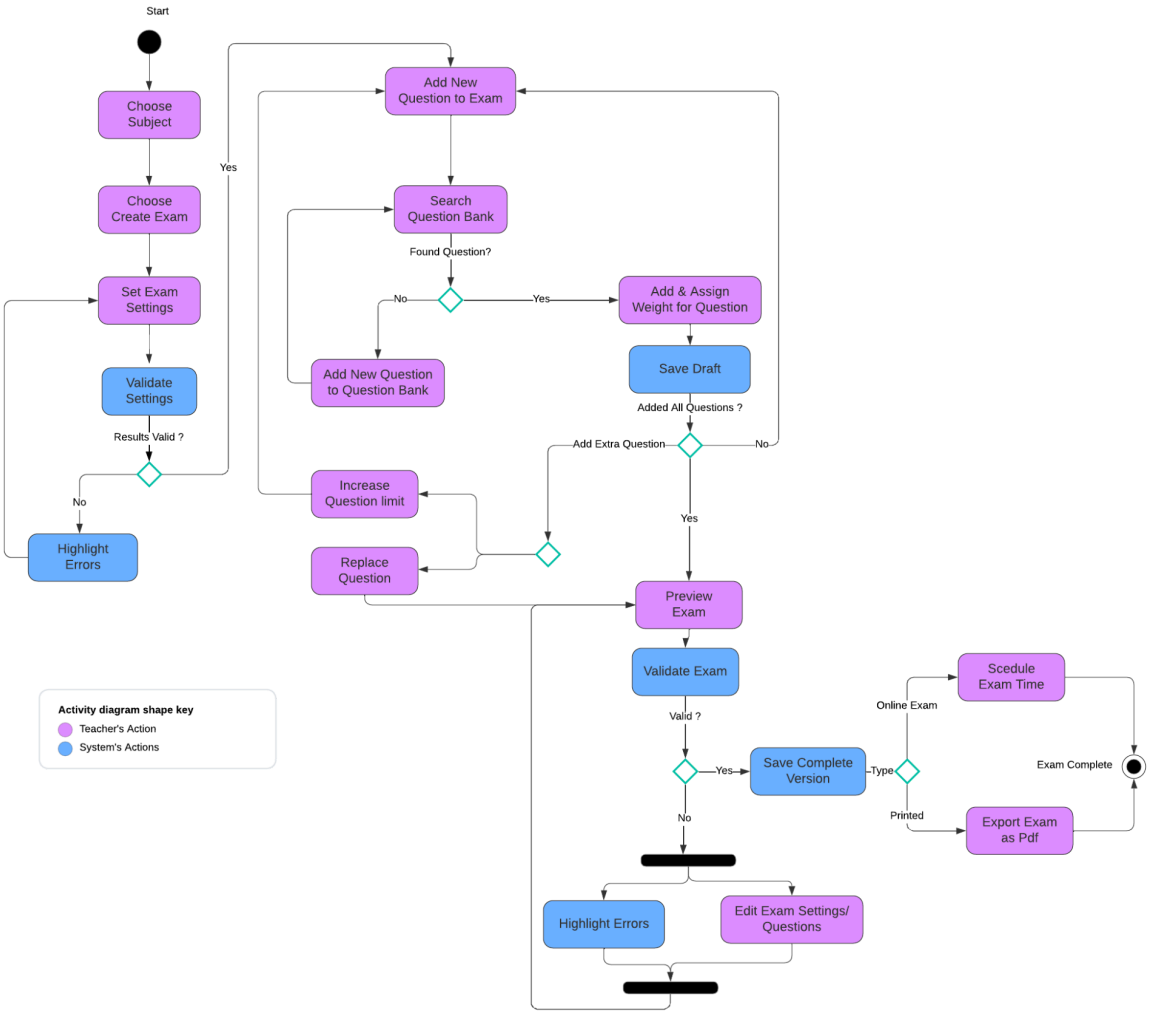
7. Extension Points

7.1 The system should allow attendance reports to be generated by teachers or principals for analysis.

7.2 The system can integrate with external systems for further analytics or compliance requirements.

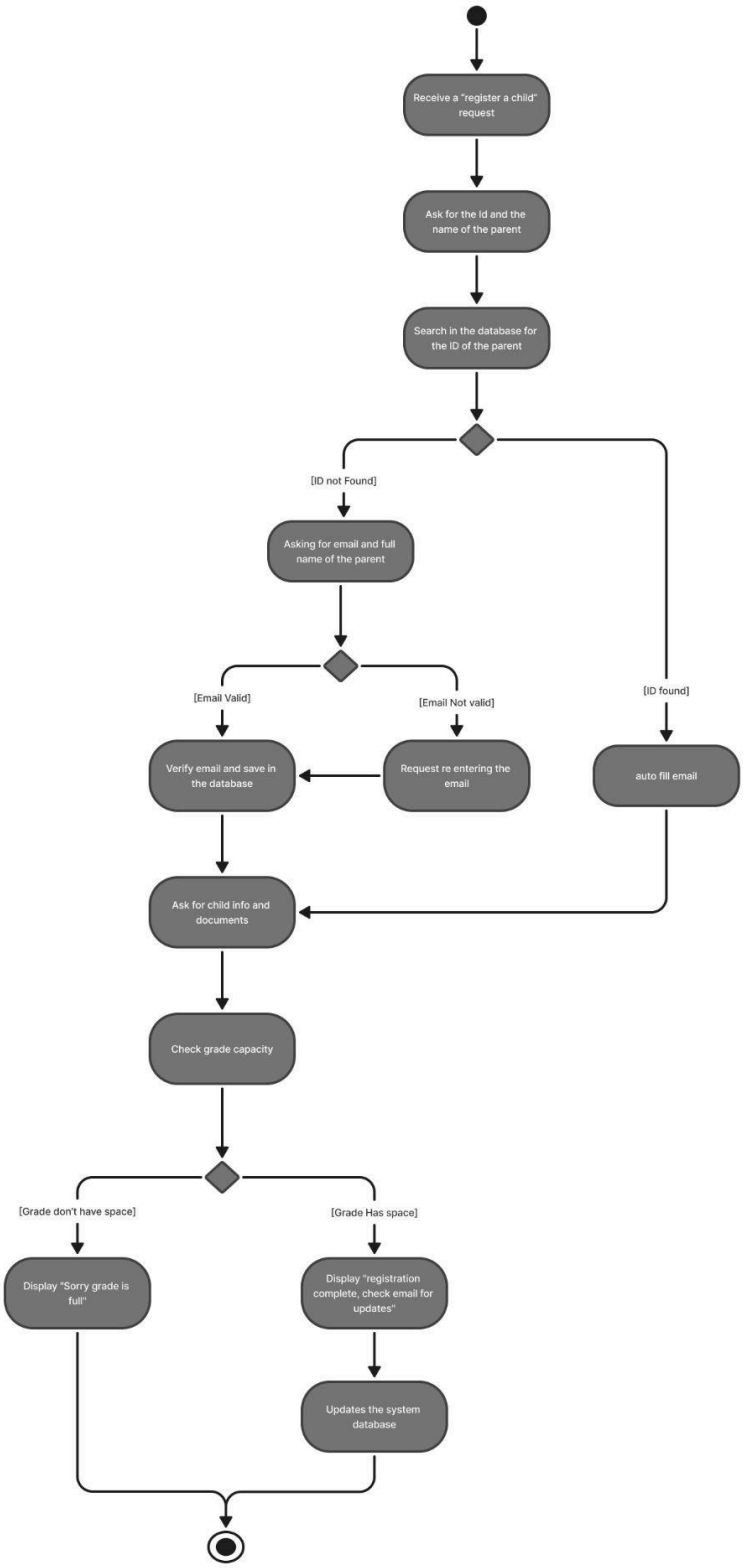
Instance Activity Diagrams

1. Create New Exam Activity Diagram (Sarah Hassouneh)[2]

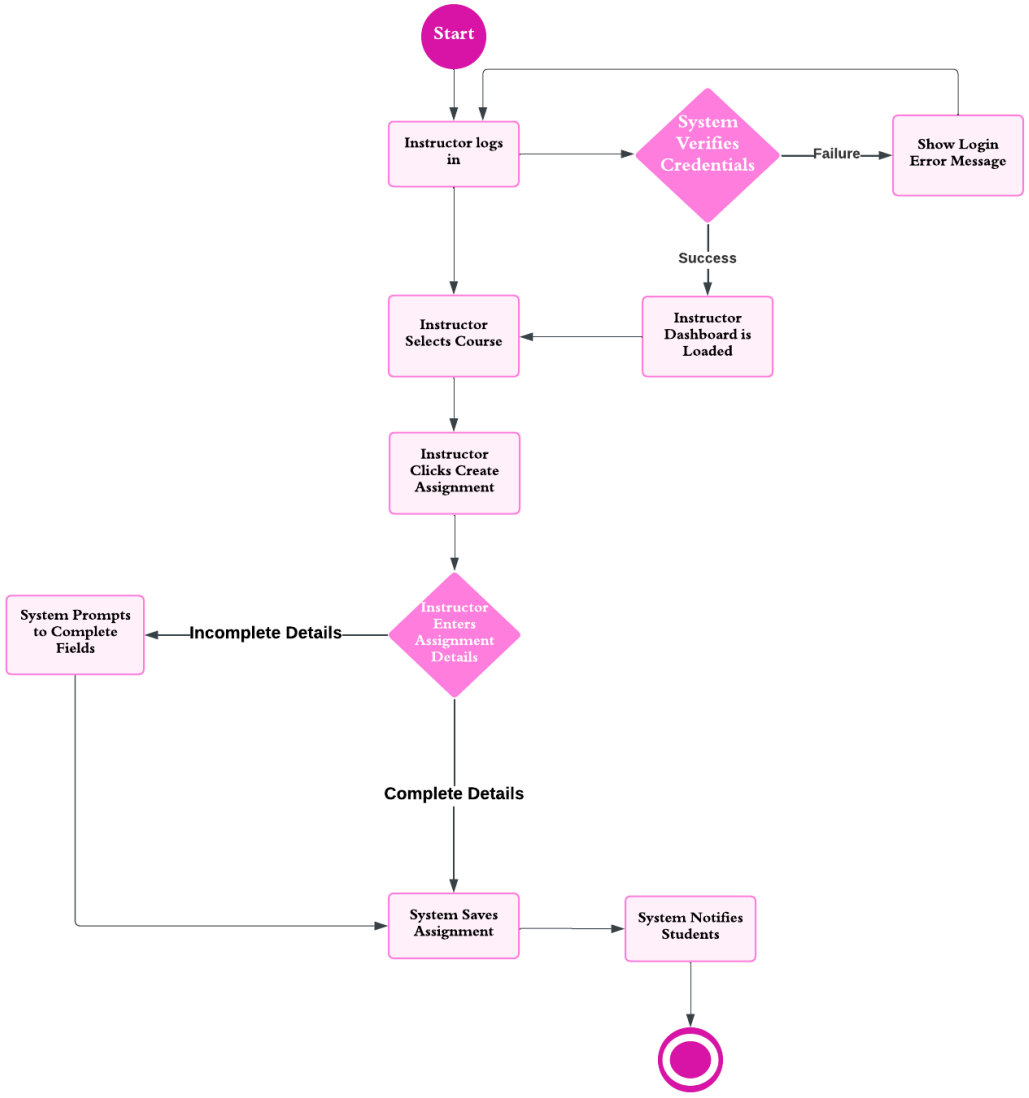


[2] View on LucidChart

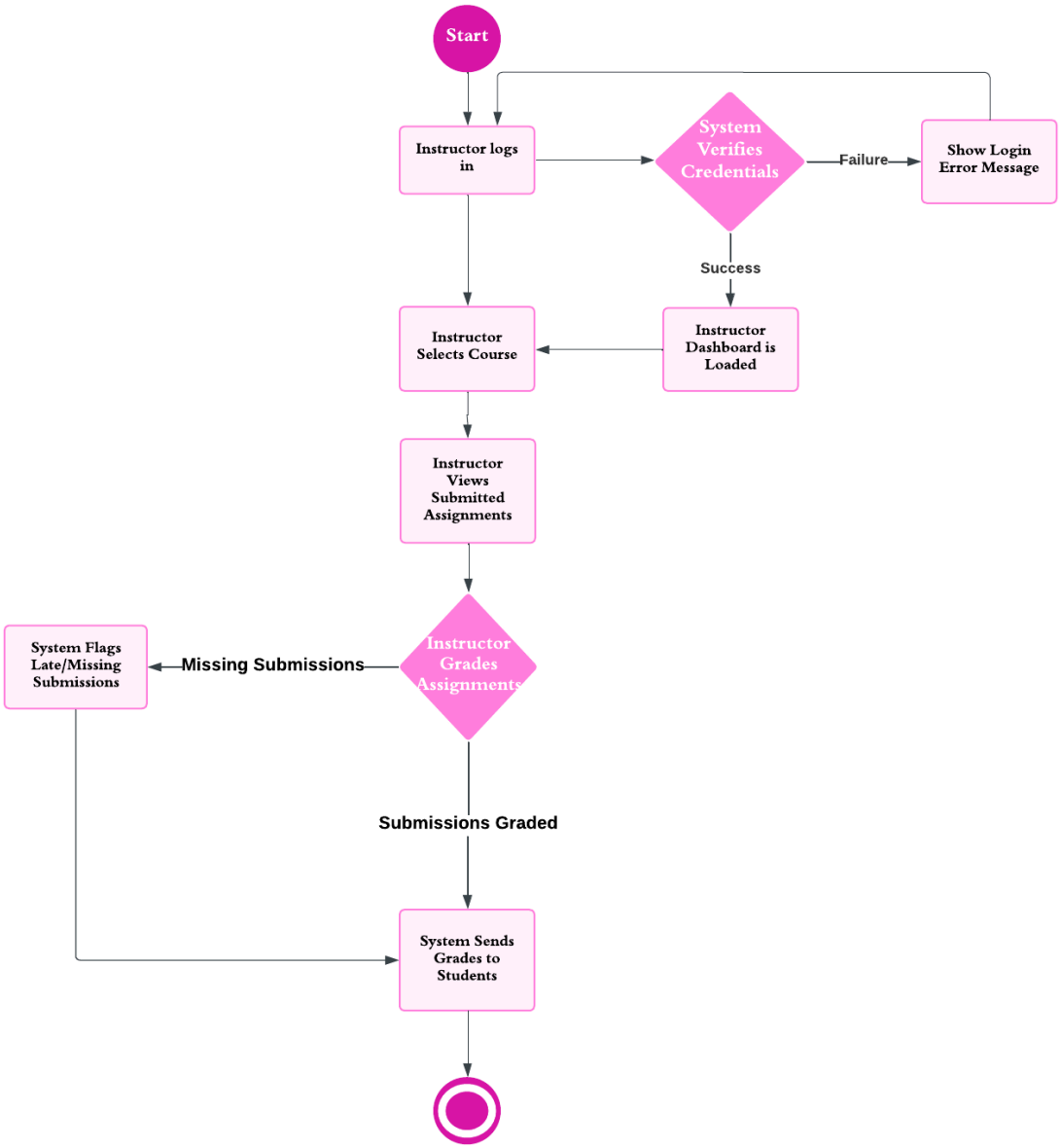
2. Register a Child Activity Diagram (Dana Hafitha)



3. Assign and Mark Assignments Activity Diagrams (Doaa Hatu)

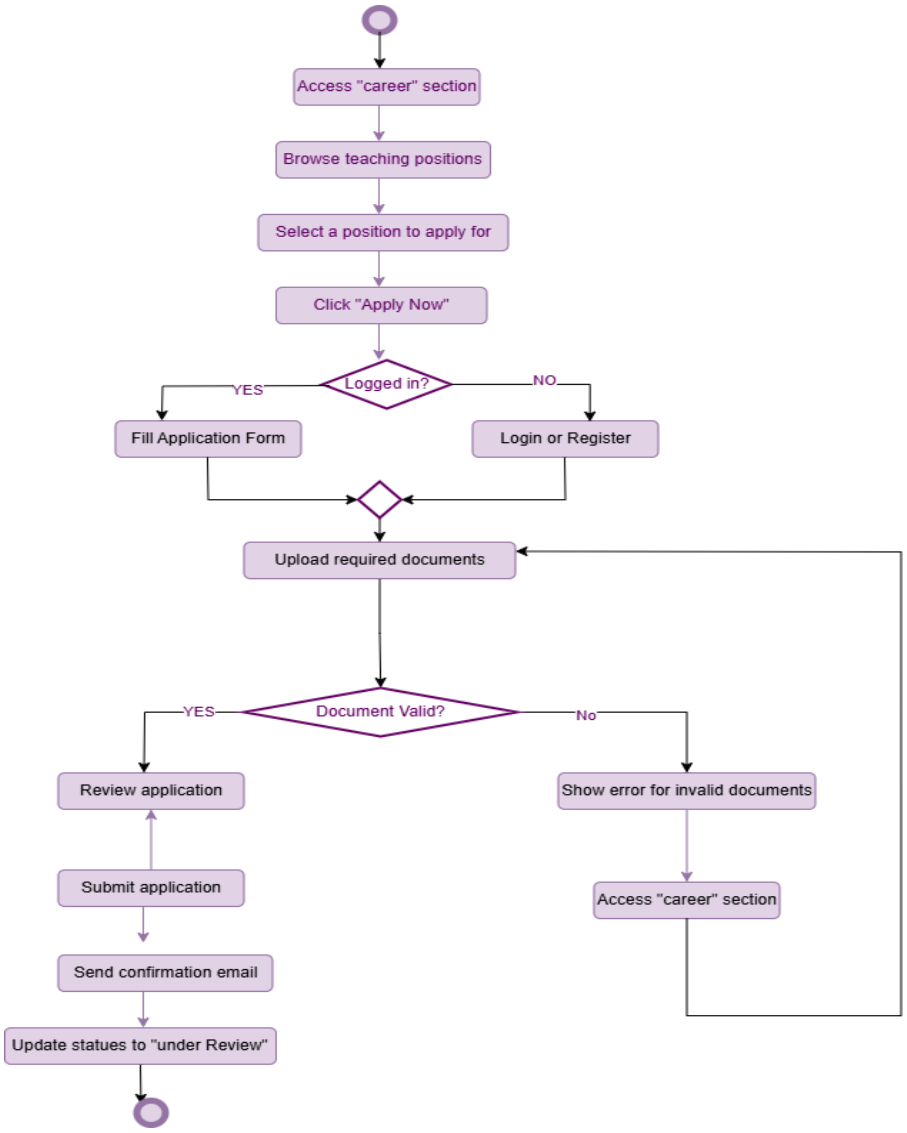


Mark Assignments



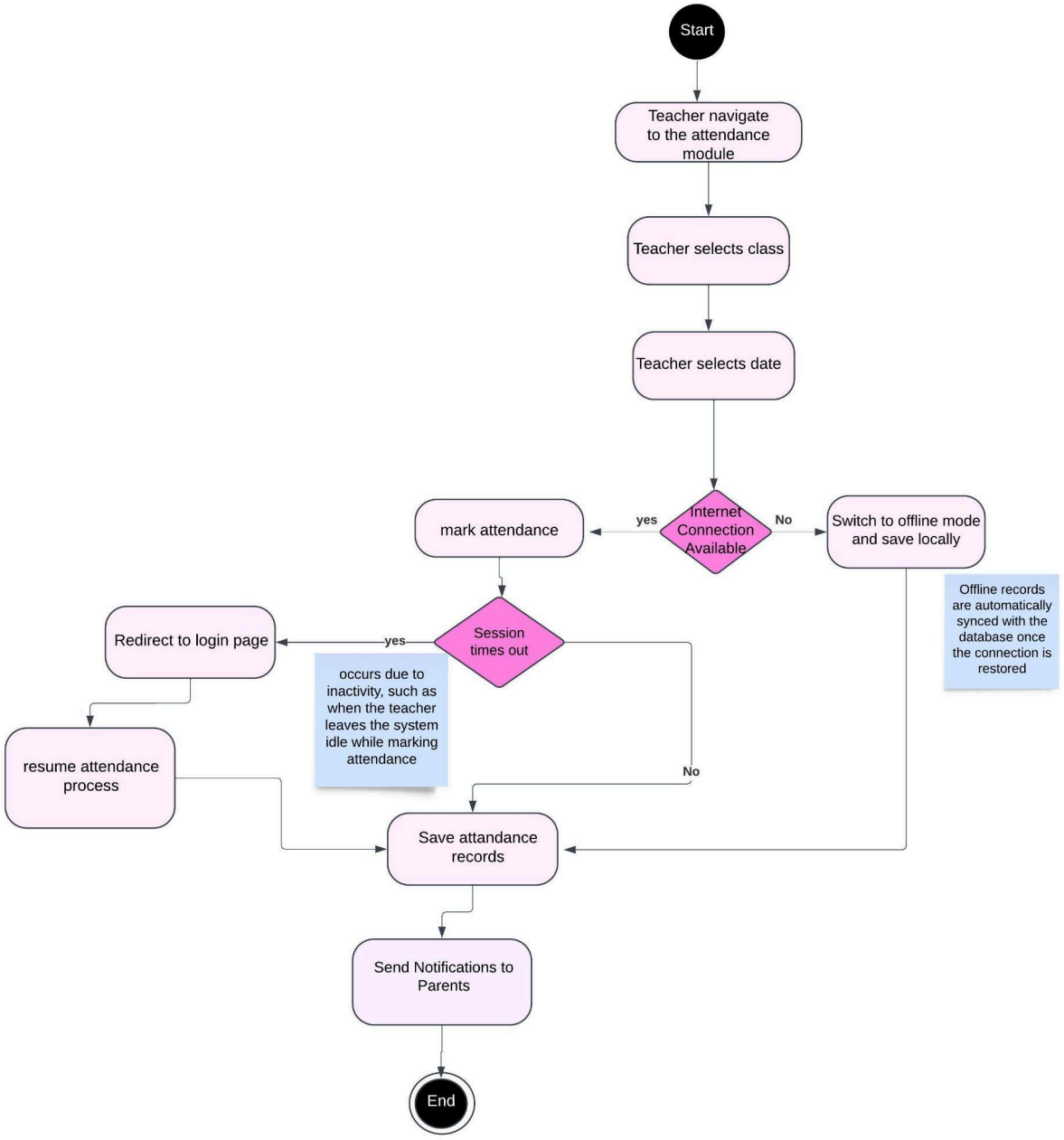
https://lucid.app/lucidchart/df3c8dfc-041d-4e36-a1b7-8320d43693b2/edit?viewport_loc=-799%2C-587%2C3219%2C1728%2C0_0&invitationId=inv_ef701df0-a9de-41ae-b49a-3b243ff3b06b **you can view my designs here :)**

4. Submit an Application for a Teaching Position Activity Diagram (Diana Naseer)



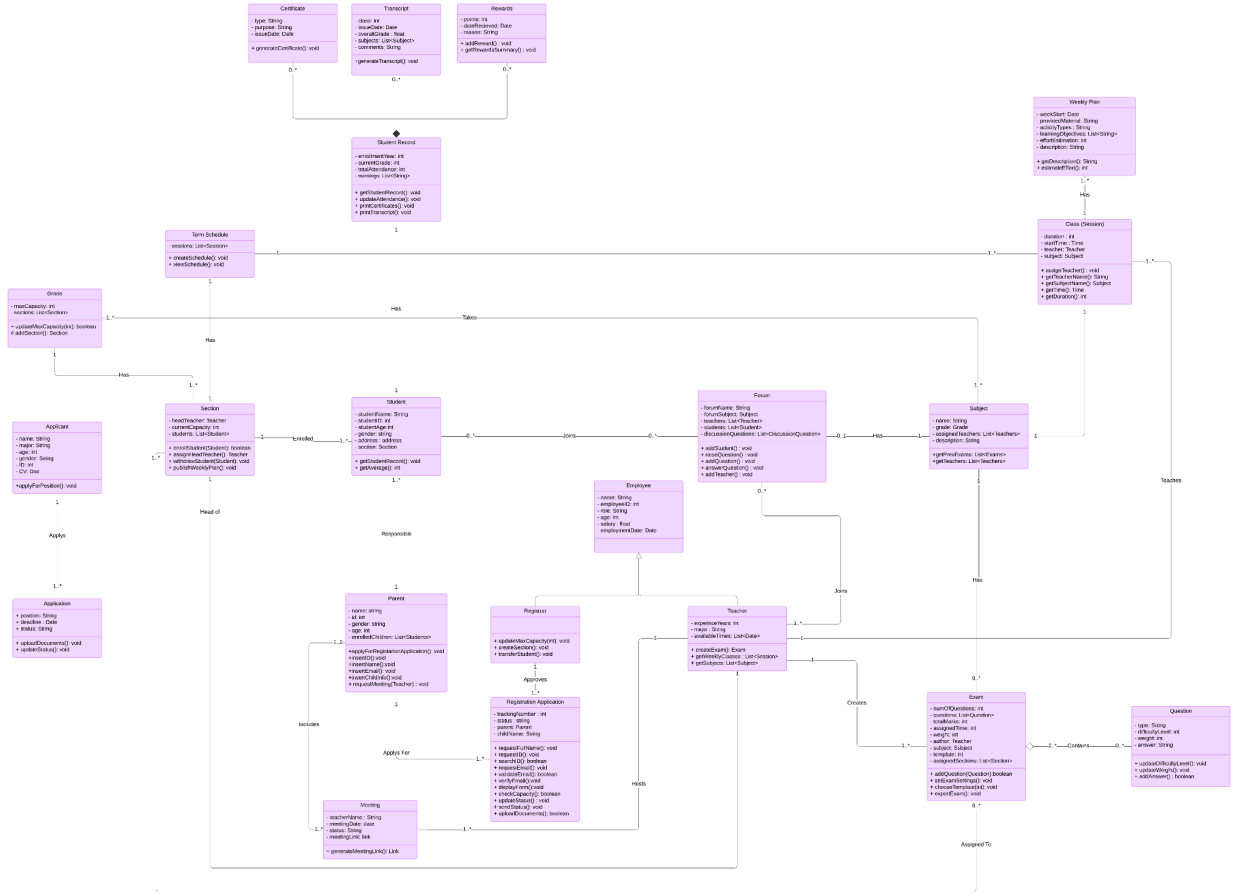
5. Update student Attendance Activity Diagram (Leen Daraghmeh)

https://lucid.app/lucidchart/6be00ac7-3e29-4d79-91aa-1b6576f1640a/edit?invitationId=inv_3bb7bad9-e098-49a1-ab3a-6a1196774581&page=0_0#



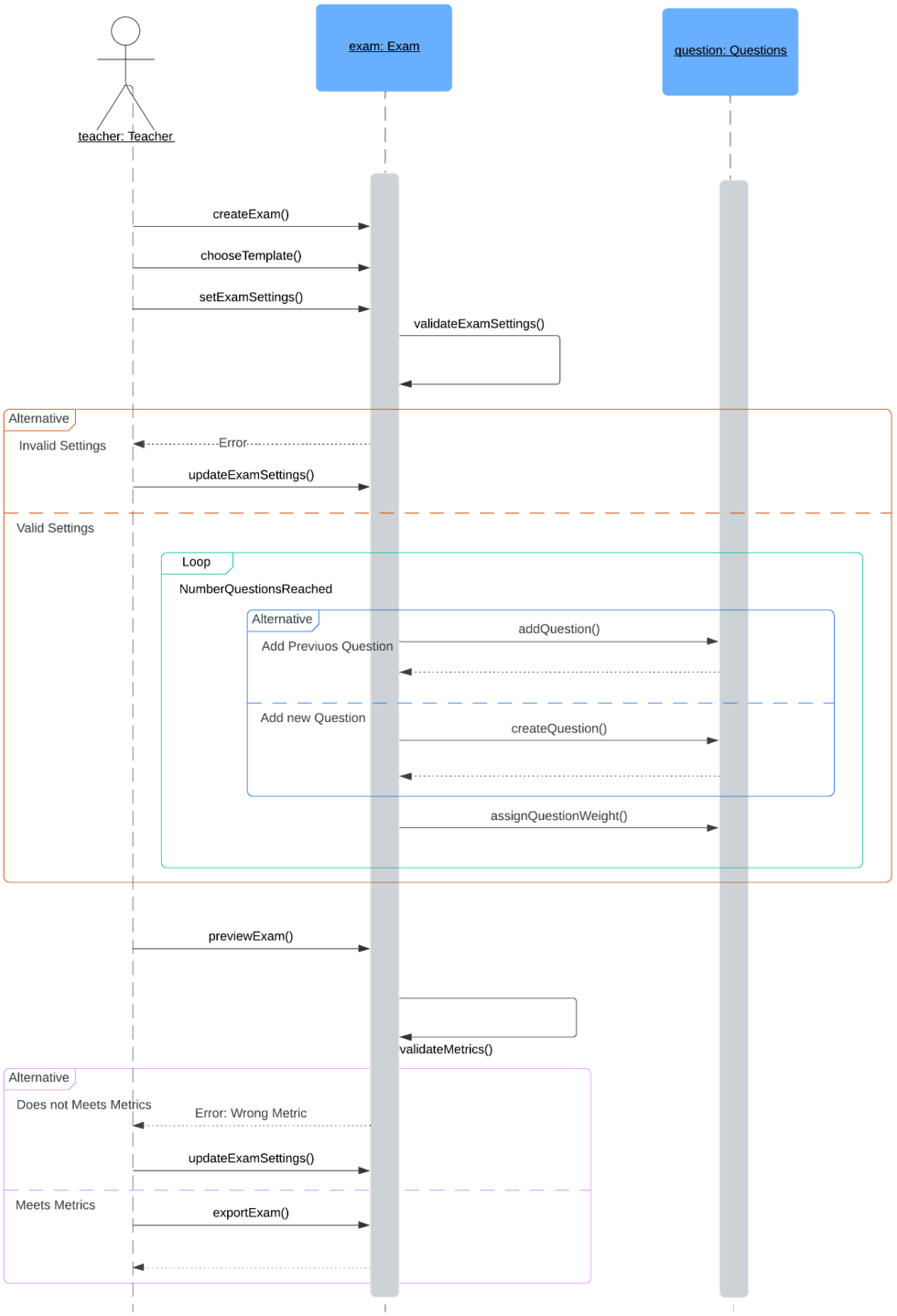
System Class Modelling

view on LucidChart:
https://lucid.app/lucidchart/7d40e9f4-e2c9-4760-91a9-415dfb60e4c3/edit?invitationId=inv_5edd7c1d-a9c1-4185-bdbc-81c646bdd75e&page=HWEp-vi-RSFO#

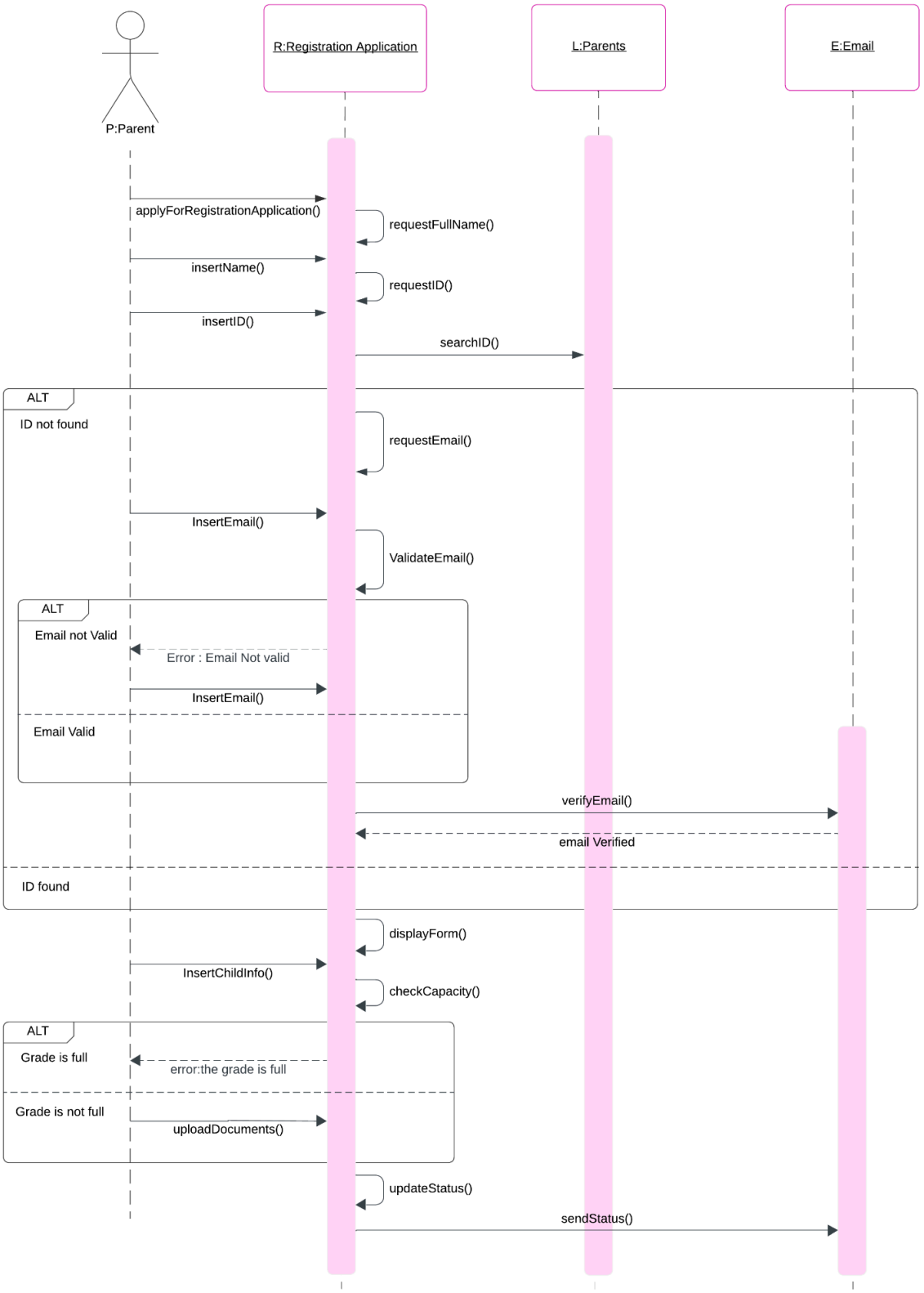

Certificate
- type: String
- purpose: String
- issueDate: Date
+ generateCertificate(): void
Transcript
- class: int
- issueDate: Date
- overallGrade : float
- subjects: List<Subject>
- comments: String
+generateTranscript(): void
Rewards
- points: int
- dateReceived: Date
- reason: String
+ addReward(): void
+ getRewardsSummary() : void
Student Record
- enrollmentYear: int
- currentGrade: int
- totalAttendance: int
- warnings: List<String>
+ getStudentRecord(): void
+ updateAttendance(): void
+ printCertificate(): void
+ printTranscript(): void
Weekly Plan
- weekStart: Date
- provisionMaterial : String
- activityTypes : String
- learningObjectives: List<String>
- differentiation: String
- description: String
+ getDescription(): String
+ estimateEffort(): int
Has
Term Schedule
- sessions: List<Session>
+ createSchedule(): void
+ viewSchedule(): void
Class (Session)
- duration : int
- startTime : Time
- teacher: Teacher
- subject: Subject
+ assignTeacher() : void
+ getTeacherName(): String
+ getSubjectName(): Subject
+ getTime(): Time
+ getDuration(): int
Grade
- maxCapacity: int
- sections: List<Section>
+ updateMaxCapacity(int): boolean
# addSection(): Section
Has
Takes
Has
Has
Applicant
- name: String
- major: String
- age: int
- gender: String
- ID: int
- CV: Doc
+applyForPosition(): void
Section
- headTeacher: Teacher
- currentCapacity: int
- students: List<Student>
+ enrollStudent(Student): boolean
+ assignHeadTeacher() : Teacher
+ withdrawStudent(Student): void
+ publishWeeklyPlan(): void
Enrolled
Student
- studentName: String
- studentID: int
- studentAge:int
- gender: string
- address : address
- section: Section
+ getStudentRecord(): void
+ getAverage(): int
Joins
Forum
- forumName: String
- forumSubject: Subject
- moderators: List<Teacher>
- students: List<Student>
- discussionQuestions: List<DiscussionQuestion>
+ addStudent() : void
+ removeStudent() : void
+ addQuestion() : void
+ answerQuestion() : void
+ addTeacher() : void
Has
Subject
- name: String
- grade: Grade
- assigned teachers: List<teachers>
- description: String
+getPrevExams: List<Exams>
+getTeachers: List<teachers>
Teaches
Head of
Responsible
Employee
- name: String
- employeeID: int
- role: String
- age: int
- salary : float
employmentDate: Date
Joins
Has
Application
+ position: String
+ deadline : Date
+ status: String
+ uploadDocuments(): void
+ updateStatus(): void
Parent
- name: string
- id: int
- gender : string
- age: int
- enrolledChildren: List<Students>
+applyForRegistration(Application): void
+insertID(): void
+insertName(): void
+insertEmail(): void
+insertChildInfo(): void
+ requestMeeting(Teacher) : void
Registrar
+ updateMaxCapacity(int): void
+ createSection(): void
+ transferStudent(): void
Approves
Teacher
- expeirenceYears: int
- major : String
- availableTimes: List<Date>
+ createExam(): Exam
+ getWeeklyClasses : List<Session>
+ getSubjects: List<Subject>
Includes
Applys For
Registration Application
- trackingNumber : int
- status : string
- parent: Parent
- childName: String
+ requestFullName(): void
+ requestID(): void
+ searchID(): boolean
+ requestEmail(): void
+ validateEmail(): boolean
+ verifyEmail(): void
+ displayForm():void
+ checkCapacity(): boolean
+ updateStatus() : void
+ sendStatus(): void
+ uploadDocuments(): boolean
Creates
Exam
- numOfQuestions: int
- questions: List<Question>
- subjects: str
- assignedTime: int
- weight: int
- author: Teacher
- subject: Subject
- template: int
- assignedSections: l ist<Section>
+ addQuestion(Question): boolean
+ initExamSettings(): void
+ chooseTemplate(int): void
+ exportExam(): void
Contains
Question
- type: Sting
- difficulty Level: int
- weight: int
- answer: String
+ updateDifficultyLevel(): void
+ updateWeight(): void
+ addAnswer() : boolean
Meeting
- teacherName : String
- meetingDate: date
- status: String
- meeting Link: link
+ generateMeetingLink(): Link
Needs
Assigned To

System Sequence Modelling

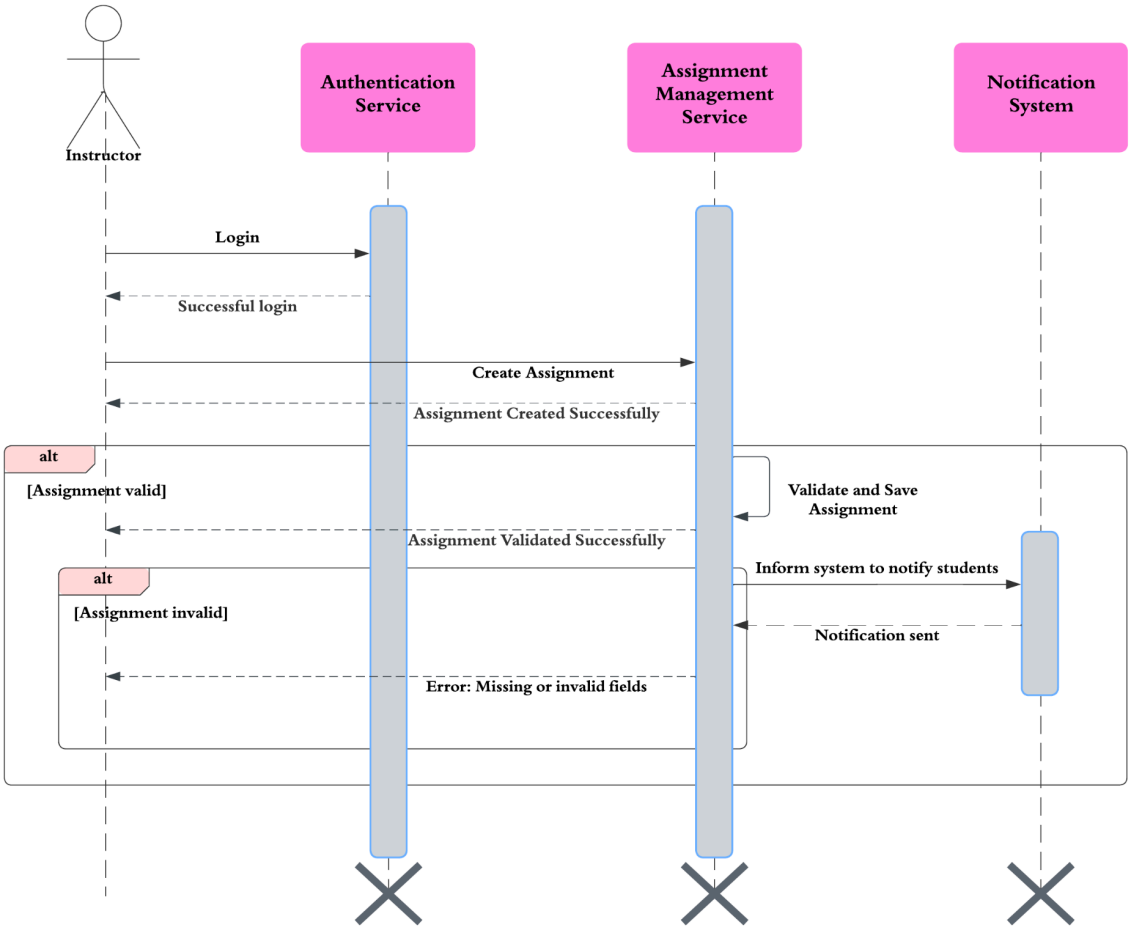
1. Use-Case: Create New Exam (Sarah Hassouneh)



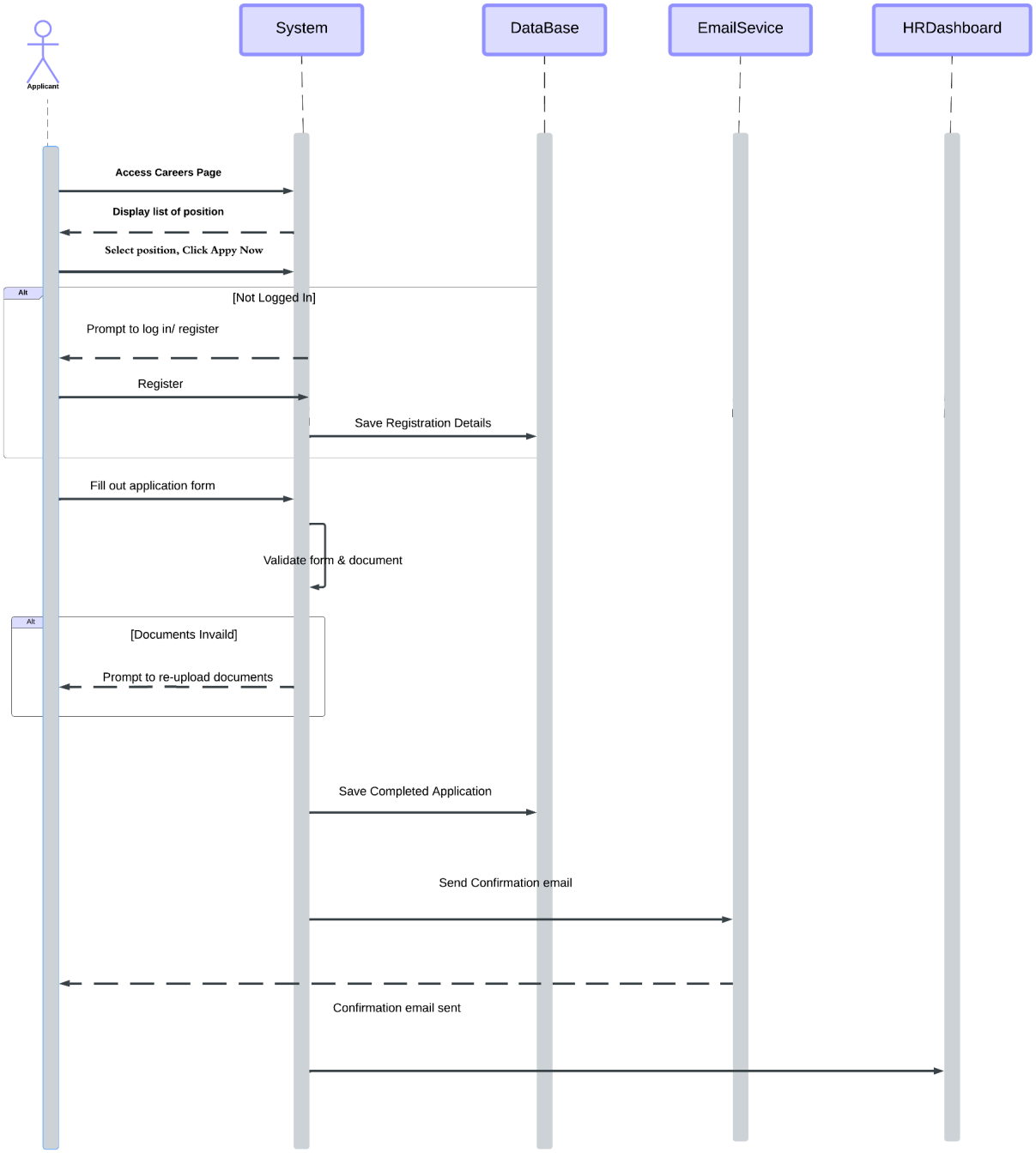
2. Register a Child Activity Diagram (Dana Hafitha)



3. Assign Assignments Activity Diagram (Doaa Hatu)

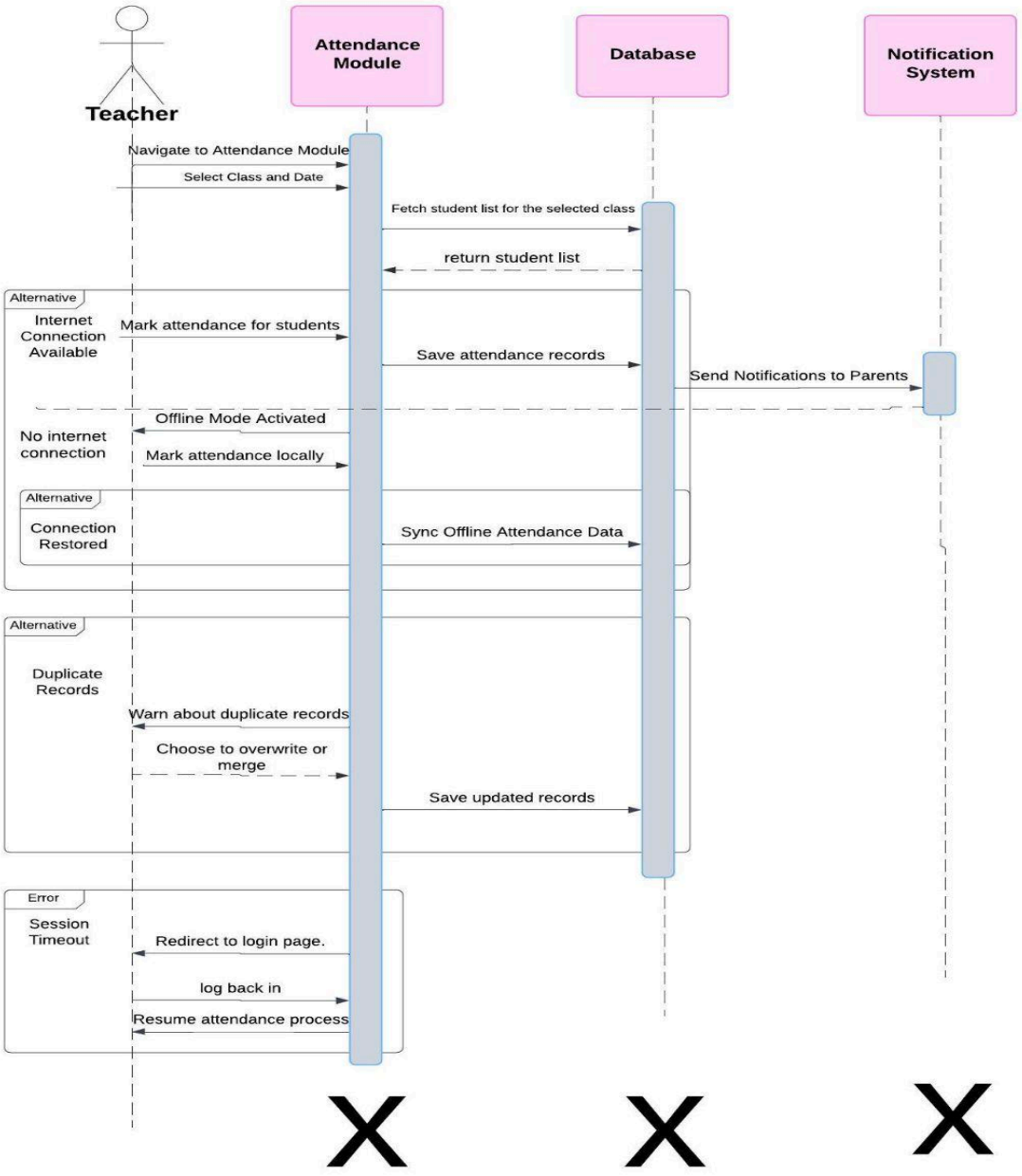


4. Submit an Application for a Teaching Position Activity Diagram (Diana Naseer)



5. Update student Attendance Activity Diagram (Leen Daraghmeh)

https://lucid.app/lucidchart/fc3173eb-c004-4aba-8c7a-cbe59a08d951/edit?beaconFlowId=9DCE5D6739E29574&invitationId=inv_730d6df7-ac72-4236-8ab6-755bf6b9ed0c&page=0_0#



System Design Goals

1- High Cohesion

The Edusmart System follows the principle of high cohesion by making related functionalities associated and grouped together into specialized modules. This supports clarity, efficiency, and easier debugging by keeping interdependent classes and functions within the same layer or component.

- **Attendance Module:**
  Dedicated to handling all attendance-related tasks, including:
  - Marking attendance.
  - Saving attendance records.
  - Syncing offline data.
    All interdependent classes, such as `AttendanceRecord` and `OfflineAttendanceStorage`, and related functions reside in this module to ensure comprehensive attendance management.
- **Notification System:**
  Focused solely on communication-related tasks, such as:
  - Sending updates to parents.
  - Automating alerts for exams, deadlines, and events.
    Classes like `ParentNotification` and `NotificationSender` are centralized within this system, consolidating all notification responsibilities to improve maintainability and scalability.

Benefits:

1- Simplifies testing, debugging, and updates.

2- Reduces cross-dependencies, ensuring smoother system development.

2- Low Coupling

The **EduSmart System** adopts the principle of low coupling by reducing interdependencies among components. It uses well-defined interfaces to ensure that changes in one component minimally impact others.

Three-Layer Architecture:

1. **User Interface (UI Layer):**
    - Handles all user interactions, such as registering students, marking attendance, or scheduling meetings.
    - Examples: Marking attendance, viewing student grades, accessing reports.
2. **Business Logic Layer:**
    - Manages core functionalities, including:

        1- Attendance processing.

        2- Notification generation.

        3- Exam creation and validation.

3. **Data Layer:**
   Handles:

        1- Data storage and retrieval.

        2- Offline data synchronization.

        3- Database management.

Encapsulation:

- Each layer focuses on its specific functionality.
- Communication between components, like the Attendance Module and Notification System, occurs via service-oriented interfaces.

Benefits:

1- Flexibility: Components can evolve independently (e.g., database updates won't affect the UI).

2- Scalability: The system can adapt to handle increasing data loads.

3- Maintainability: Minimizes cascading errors across layers.

3- High Security

The system prioritizes the protection of sensitive user data (e.g., teacher credentials, student records, payment details).

Security Features:

1. **Separate Servers:** Application logic and database operations run on separate servers to enhance security.
2. **Role-Based Access Control (RBAC):**
    - Teachers can mark attendance but cannot modify financial records.
    - Parents can only view their child’s progress and not access other students' data.
3. **Data Encryption:**
    - All sensitive data (e.g., passwords, financial details) is encrypted.
4. **Regular Backups:**
    - Ensures data recovery in case of breaches or failures.

Benefits:

1- Protects against unauthorized access.

2- Ensures data integrity and confidentiality.

3- Builds trust among system users.

4- Performance Optimization

The system is designed for efficiency, ensuring that critical operations are completed quickly, even during high-load scenarios.

Optimizations:

1. **Offline Attendance Syncing:**
    - Ensures syncing of offline attendance data is completed within 3-5 seconds of connection restoration.
2. **Asynchronous Notification Delivery:**
    - Notifications (e.g., absence alerts) are sent within 5 minutes using background processes.

Benefits:

1- Faster response times.

2- Improved user satisfaction.

3- Reliable performance during peak usage (registration periods or examination results).

5- Scalability

The system is designed to accommodate growing user bases, ensuring reliability and performance as the school expands.

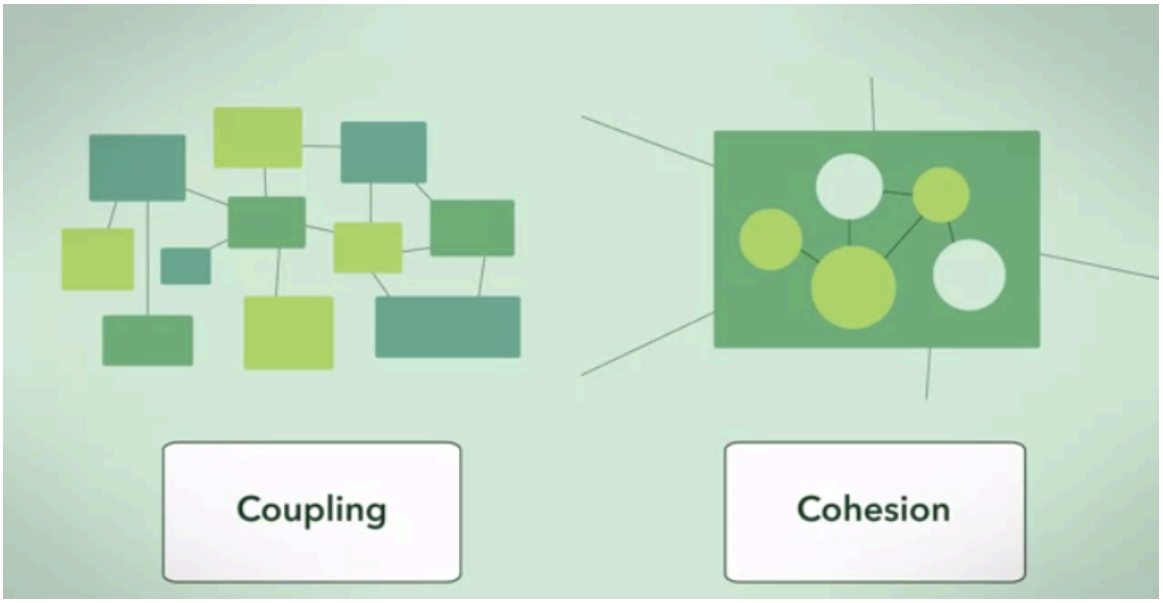
Scalable Features:

1. **Load Balancers:**
    - Distribute traffic across multiple servers to avoid bottlenecks.
2. **Scalable Database Architecture:**
    - Use sharding or indexing to efficiently manage large datasets.

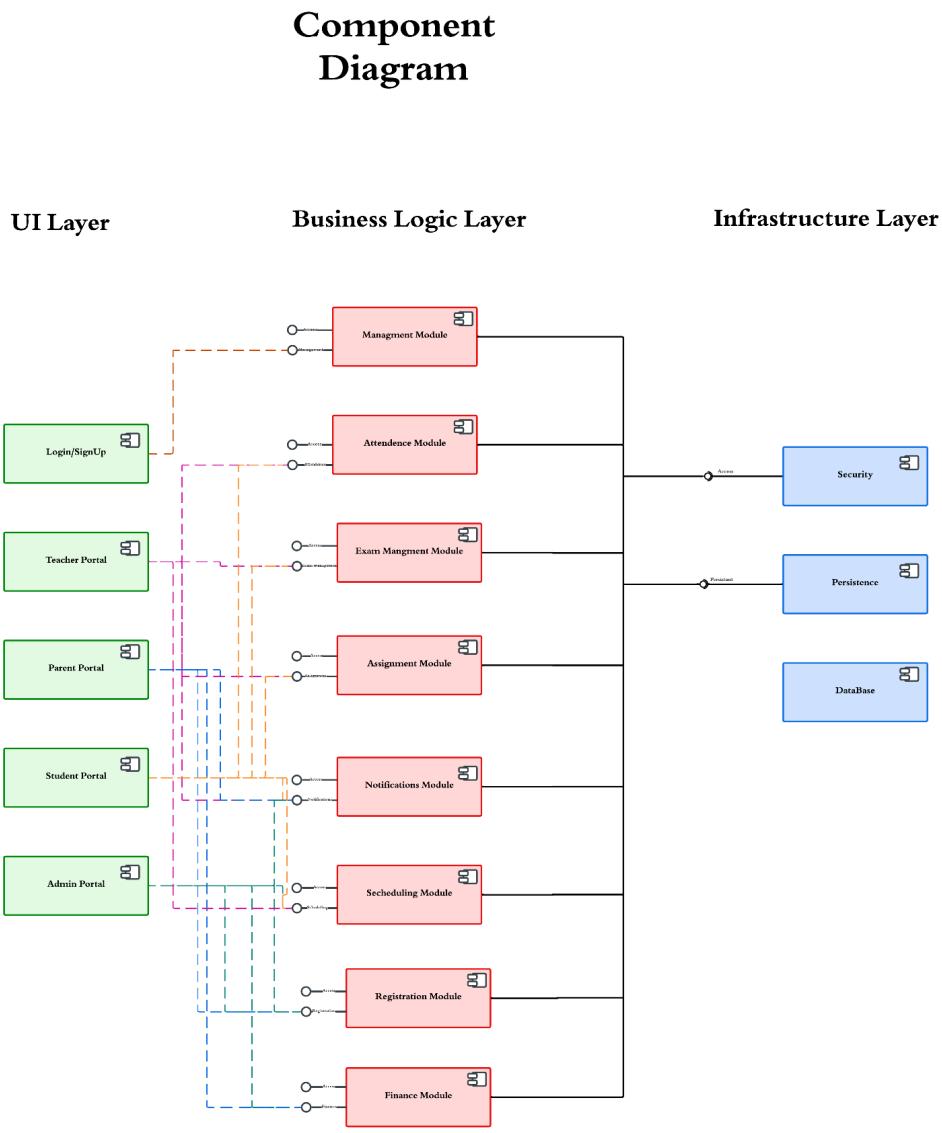
Benefits:

1-Supports growth without degrading performance.

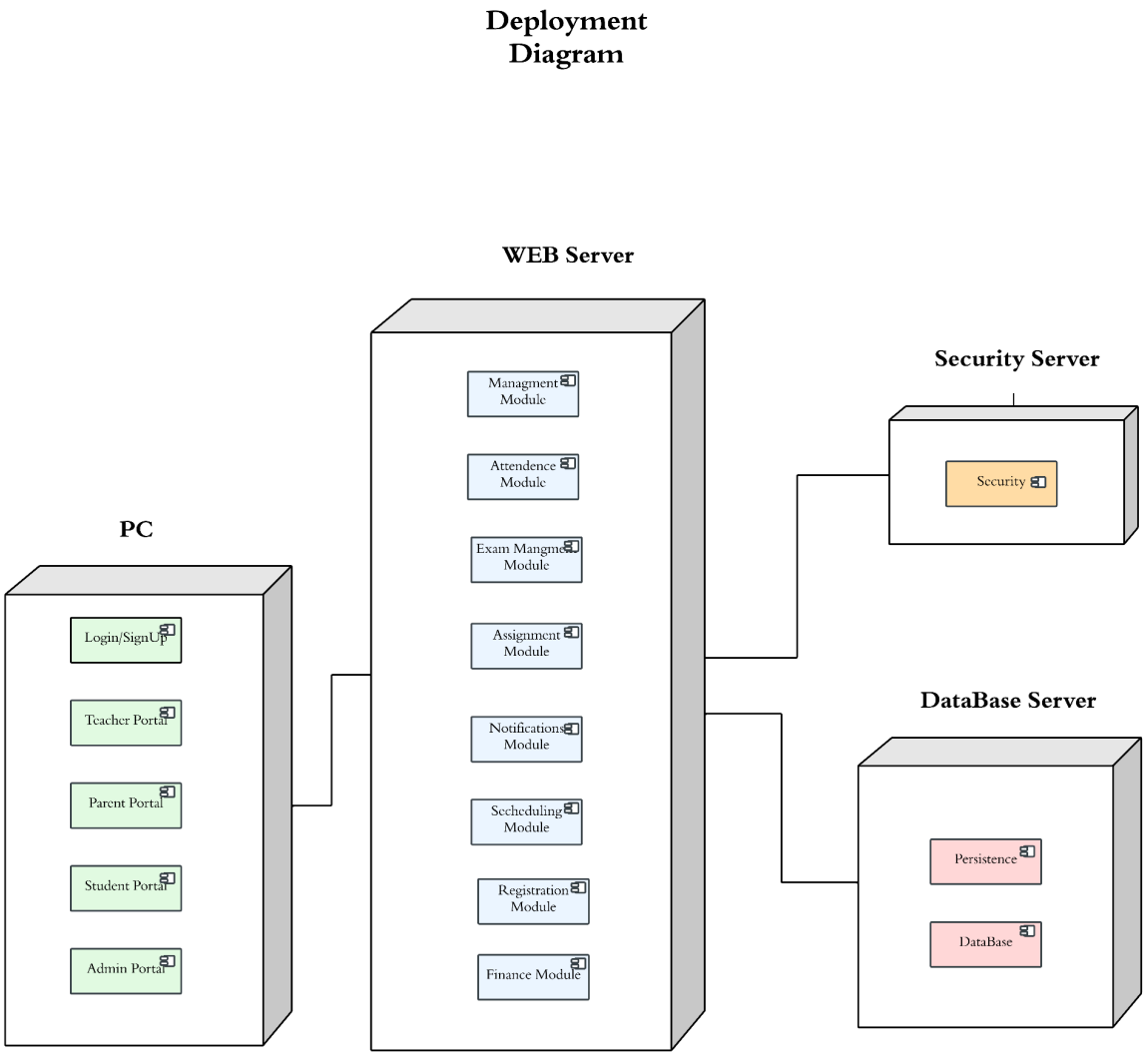
2- Handle increased workloads (more students, parents, and teachers).



System Component Design



System Deployment Design



view component diagram and deployment diagram on lucidchart: https://lucid.app/lucidchart/bdec23b1-142b-4639-b901-b5f7da18e2e5/edit?viewport_loc=-1186%2C-2190%2C5575%2C4094%2C0_0&invitationId=inv_ab004367-fc12-40dd-a5d4-81c0de0f2a04