

PM3: Project final report

Project Name: "Eduzest"

Team Members:

Name	Email	ID
Mohamed Mohamed Ibrahim Al sariti	mohamed.alsariti.2023@aiu.edu.eg	22101901
Ahmed Adel Abdelrahman Ibrahim	ahmed.ibrahim.2023@aiu.edu.eg	22101163
Merna Adel Abdelrahman Ibrahim	merna.ibrahim.2023@aiu.edu.eg	22101164
Nadira Mohamed abdallah Ismail Elsirafy	nadira.abdallah.2023@Aiu.edu.eg	22101377
Mariam elrafei mohamed	Mariam.abdelfttah.2023@Aiu.edu.eg	22101442

Description:

"Eduzest" is an innovative online exam application designed to streamline the exam taking experience for students. With features like **QR paper scanning (future not require now)** and AI-based grading, "Eduzest" ensures a seamless and efficient process for students to take exams and receive instant results.

In recent years, it has become clear that many students in the secondary stage face a problem in tests and training for final exams. Therefore, we had to take this into consideration and created "Eduzest" as an educational application that provides a suitable environment for the student to take exams under pressure of time, which makes the simulation more realistic, such as: Final exams, and also with the continuous progress in artificial intelligence, we had to add it to our application to be one of our new teachers in correcting exams accurately, Which made us faster in correcting exams with high accuracy and sending the model answer to the student .

We also did not forget the importance of communication between the student and his teacher, so we created a communication page between them. We also compared the student's grade with his classmates and sent an alert notification if the grade was not good, and we added a ranking page for the first places.

Customer Requirement:

Exam Access and Dashboard

User-friendly dashboard for students to access exams and view results.

Exam Taking

Ability for students to take exams online through the mobile app.

QR Paper-Based Exams (future not require now)

Option for students to solve exams on physical paper and scan their answers using the mobile app.

AI-Based Grading

Automatic grading of exam papers using advanced AI algorithms.

Instant Result Notification

Instant notification to students with their exam results.

Ranking

Provide students with a ranking or comparison of their performance with others.

Subsystems and Functions:

Authentication Subsystem

Responsible for managing user authentication and ensuring secure sign-up and login and manage profile functionality

Exam Manager Subsystem

Facilitates the process of taking exams online and scanning QR codes for paper-based exams and add exam and delete exam and edit exam and submit exam.

Grading Manager Subsystem

Publish result and display new result for student

Notification Manager Subsystem

Send notification for student and display it for student

Ranking Manager Subsystem

Compiles and presents rankings or comparisons of students' performance with others

Reporting Manager Subsystem

Make report for student marks by admin

Question Bank Subsystem

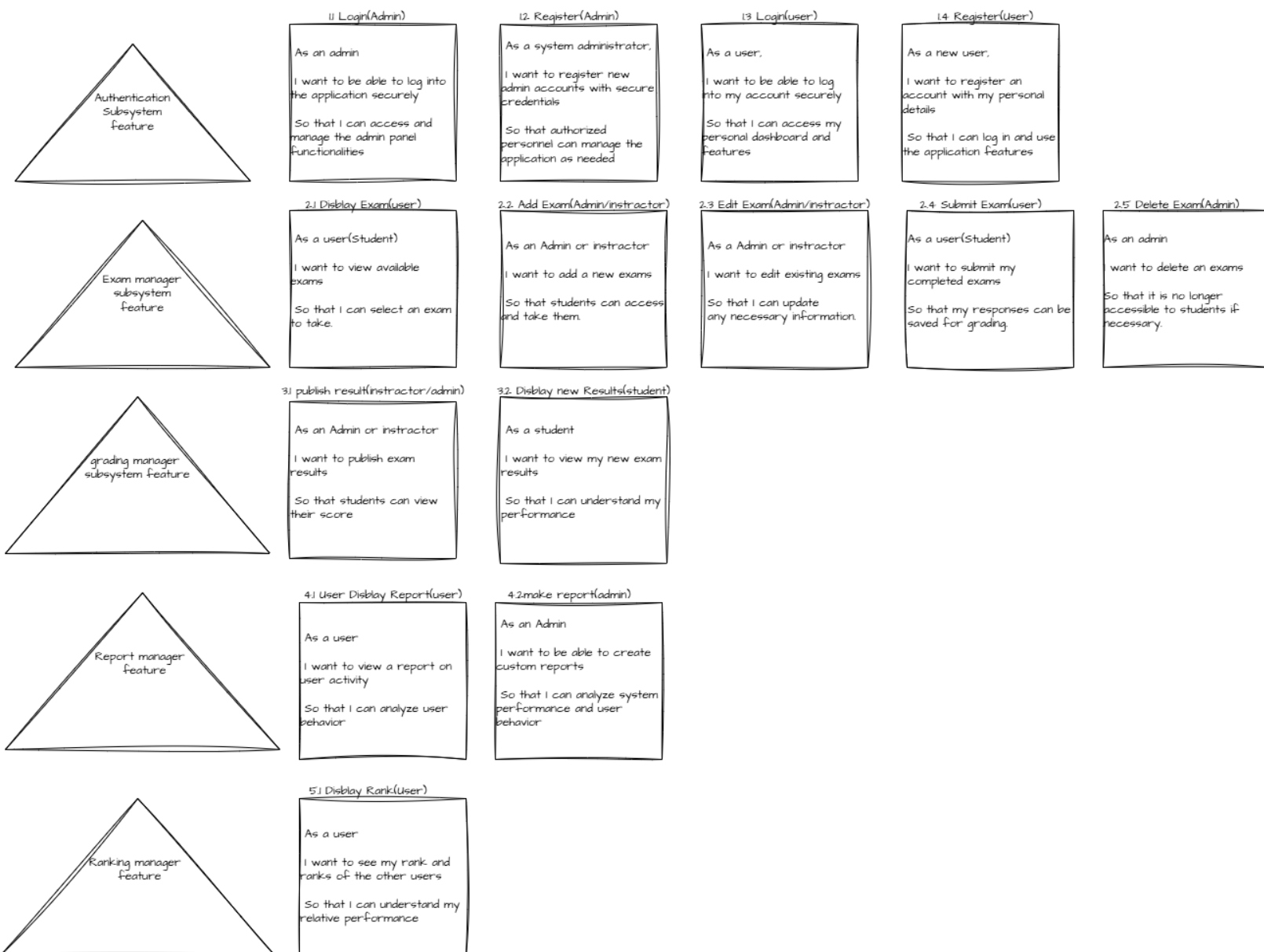
Manages a database of exam questions, including creation, editing, categorization, and retrieval and display it for student

Feedback Subsystem

Allows students to provide feedback on exams, user experience, and suggestions for improvement

1-Stories estimations, use case estimations, feature estimations. Use agile and formal methods to come up with estimates:

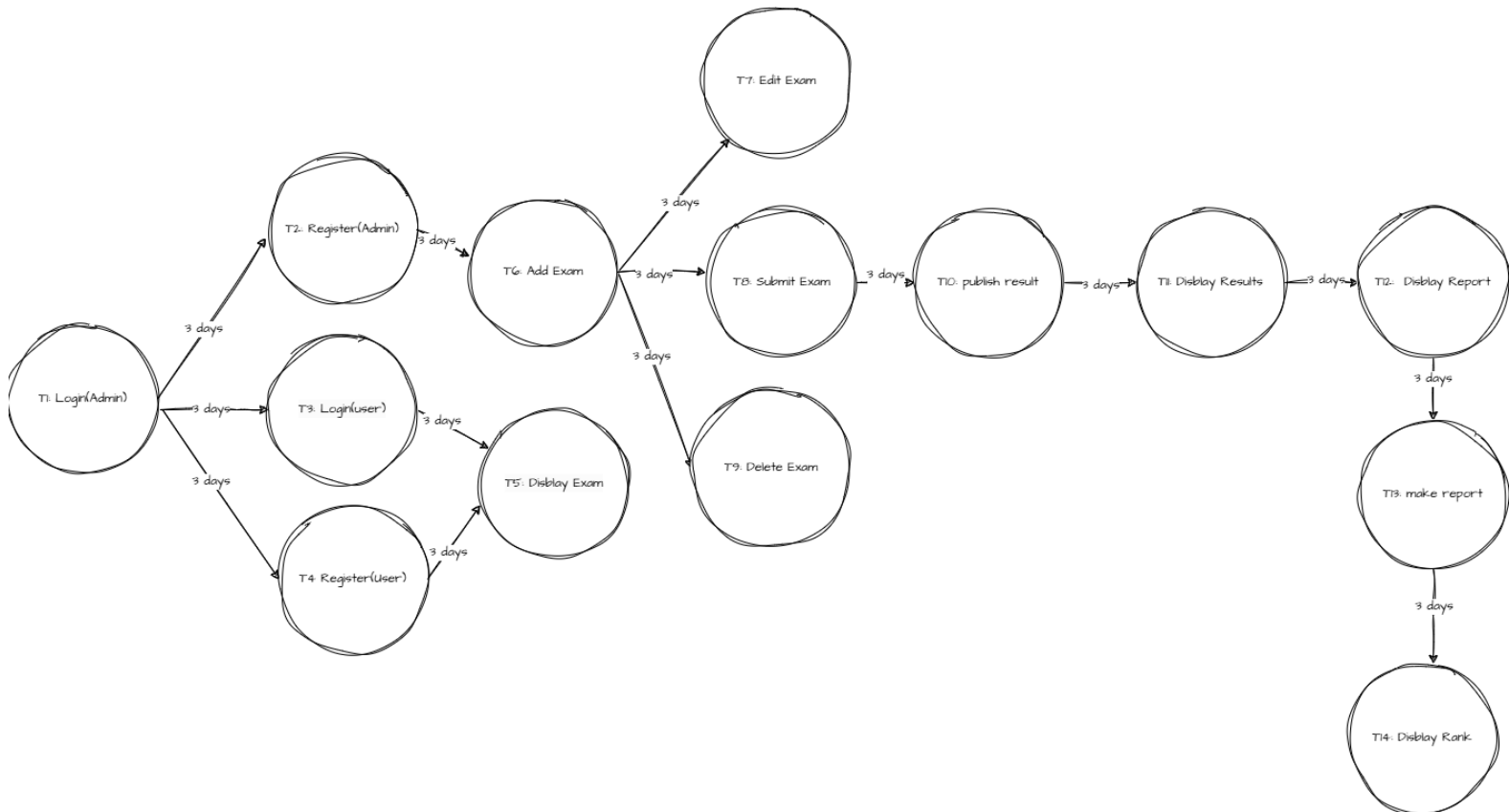
(user stories)



All Diagrams Created By [Draw.io](https://draw.io) for best view please follow link:

<https://shorturl.at/Yv5kS>

(pert chart)



This Diagram Created By **Draw.io** For Best View please Follow Link:

<https://shorturl.at/zCxm>

Planning Poker:

User Stories:

1. Authentication Subsystem

- 1.1 Login (Admin)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 1.2 Register (Admin)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 1.3 Login (User)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 1.4 Register (User)
Recommended points: $(3+3+3+3+3) / 5 = 3 \sim 3$ points
Working days: $3 * 2 = 6$ days

2. Exam Management Subsystem

- 2.1 Display Exam (User)
Recommended points: $(2+3+3+3+3) / 5 = 2.8 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 2.2 Add Exam (Admin/Instructor)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 2.3 Edit Exam (Admin/Instructor)
Recommended points: $(3+3+1+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 2.4 Submit Exam (User)
Recommended points: $(2+3+3+3+3) / 5 = 2.8 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 2.5 Delete Exam (Admin)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days

3. Grading Subsystem

- 3.1 Publish Result (Instructor/Admin)
Recommended points: $(3+3+3+3+3) / 5 = 3 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 3.2 Display New Results (Student)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days

4. Report Management Subsystem

- 4.1 User Display Report (User)
Recommended points: $(3+3+2+3+3) / 5 = 2.8 \sim 3$ points
Working days: $3 * 2 = 6$ days
- 4.2 Make Report (Admin)
Recommended points: $(3+3+3+3+3) / 5 = 3 \sim 3$ points
Working days: $3 * 2 = 6$ days

5. Ranking Subsystem

- 5.1 Display Rank (User)
Recommended points: $(2+3+2+3+3) / 5 = 2.6 \sim 3$ points
Working days: $3 * 2 = 6$ days

Feature Estimation:

- Feature 1: Authentication Subsystem Feature
Points: 12
Working days: 34
- Feature 2: Exam Manager Subsystem Feature
Points: 15
Working days: 30
- Feature 3: Grading Manager Subsystem Feature
Points: 6
Working days: 12
- Feature 4: Report Manager Feature
Points: 6
Working days: 12
- Feature 5: Ranking Manager Feature
Points: 3
Working days: 6

Total:

- **Total Story Points: 42 points**
- **Total Working Days: 84 days**

2-A schedule for different tasks and activities in the project, indicating the different milestones and visibility points. Use two-weeks milestones:

Phase 1: Project Initialization (Week 1-2)

Week 1

- Total points for the week: 4 points
- Tasks:
 - Gather and analyze system requirements (functional and non-functional requirements) – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 4) = 7 \text{ days}$

Week 2

- Total points for the week: 11 points
 - Tasks:
 1. Define project scope and objectives – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 11) \approx 1.9 \text{ days}$
 2. Create Use Case Diagram – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 11) \approx 1.9 \text{ days}$
 3. Identify actors, use cases, and relationships – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 11) \approx 1.9 \text{ days}$
 4. Complete initial project documentation – 2 points
 - Time allocated: $7 \text{ days} \times (2 \div 11) \approx 1.3 \text{ days}$
 - Milestone 1: Finalized system requirements and Use Case Diagram
 - Visibility Point: Stakeholder approval of requirements and project scope
-

Phase 2: System Design (Week 3-4)

Week 3

- Total points for the week: 8 points
- Tasks:
 1. Design database schema for storing exam, question, result, and user information – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 8) = 3.5 \text{ days}$

2. Create system architecture and interaction diagrams – 4 points

- Time allocated: $7 \text{ days} \times (4 \div 8) = 3.5 \text{ days}$

Week 4

- Total points for the week: 6 points
- Tasks:
 1. Develop the UI – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 6) \approx 3.5 \text{ days}$
 2. Define API endpoints for student and admin functionalities – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 6) \approx 3.5 \text{ days}$
- Milestone 2: Completion of database design and finalized UI
- Visibility Point: Approval of system design and UI/UX

Phase 3: Backend Development (Week 5-6)

Week 5

- Total points for the week: 7 points
- Tasks:
 1. Set up the backend environment (e.g., Node.js, Express, MongoDB) – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 7) \approx 3 \text{ days}$
 2. Implement authentication (register/login) functionality – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 7) \approx 4 \text{ days}$

Week 6

- Total points for the week: 9 points
- Tasks:
 1. Create APIs for admin and student actions (adding, updating, and removing questions; exam-taking process; result submission) – 5 points
 - Time allocated: $7 \text{ days} \times (5 \div 9) \approx 3.9 \text{ days}$
 2. Integrate automated system actions like notifications and ranking updates – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 9) \approx 3.1 \text{ days}$
- Milestone 3: Backend API implementation for core functionalities

- Visibility Point: API testing with Postman tool

Phase 4: Frontend Development (Week 7-8)

Week 7

- Total points for the week: 7 points
- Tasks:
 1. Set up the frontend environment (e.g., React or Angular) – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 7) \approx 3 \text{ days}$
 2. Develop dashboard pages for students and admins – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 7) \approx 4 \text{ days}$

Week 8

- Total points for the week: 7 points
- Tasks:
 1. Integrate frontend with backend APIs – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 7) \approx 4 \text{ days}$
 2. Ensure responsiveness and cross-browser compatibility – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 7) \approx 3 \text{ days}$
- Milestone 4: Completion of frontend UI with working integration
- Visibility Point: Functional demo of student and admin dashboards

Phase 5: Testing & Quality Assurance (Week 9-10)

Week 9

- Total points for the week: 8 points
- Tasks:
 1. Perform unit testing for individual modules – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 8) = 3.5 \text{ days}$
 2. Conduct system integration testing to ensure end-to-end functionality – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 8) = 3.5 \text{ days}$

Week 10

- Total points for the week: 8 points
- Tasks:
 1. Test edge cases for authentication, exam submissions, and result publications – 5 points
 - Time allocated: $7 \text{ days} \times (5 \div 8) \approx 4.4 \text{ days}$
 2. Gather user feedback through beta testing – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 8) \approx 2.6 \text{ days}$
- Milestone 5: Successful completion of testing with resolved bugs
- Visibility Point: Testing approval from stakeholders

Phase 6: Deployment and Documentation (Week 11-12)

Week 11

- Total points for the week: 7 points
- Tasks:
 1. Deploy the system on a cloud platform (e.g., AWS, Azure) – 4 points
 - Time allocated: $7 \text{ days} \times (4 \div 7) \approx 4 \text{ days}$
 2. Finalize user manuals and technical documentation – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 7) \approx 3 \text{ days}$

Week 12

- Total points for the week: 6 points
- Tasks:
 1. Conduct training sessions for admins and end-users – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 6) \approx 3.5 \text{ days}$
 2. Create a system maintenance plan – 3 points
 - Time allocated: $7 \text{ days} \times (3 \div 6) \approx 3.5 \text{ days}$
- Milestone 6: Deployment of the live system and delivery of documentation
- Visibility Point: System go-live and formal project closure

::From 20/10/2024 to 27/12/2024:

Week	Phase	Tasks	Milestone	Visibility Point	Start/End Date
1-2	Project Initialization	Requirement gathering, Use Case Diagram, initial documentation	Finalized system requirements and Use Case	Stakeholder approval of requirements	Saturday, Oct 20, 2024 to Sunday, Oct 27, 2024
3-4	System Design	Database schema, architecture design, API definitions	Database design and UI completed	Approval of system design	Monday, Oct 28, 2024 to Sunday, Nov 10, 2024
5-6	Backend Development	Backend setup, authentication, APIs for exam-related functionalities	Backend API implementation	API testing	Monday, Nov 11, 2024 to Sunday, Nov 24, 2024
7-8	Frontend Development	Frontend setup, dashboards for students and admins, API integration	Frontend UI with integration completed	Functional demo of dashboards	Monday, Nov 25, 2024 to Sunday, Dec 8, 2024
9-10	Testing & Quality Assurance	Unit testing, integration testing, beta testing, bug resolution	Successful testing phase	Testing approval	Monday, Dec 9, 2024 to Sunday, Dec 22, 2024
11-12	Deployment and Documentation	Deployment, manuals, user training, maintenance plan	System deployed and documented	System go-live	Monday, Dec 23, 2024 to Friday, Dec 27, 2024

3-A demo for one or two features that are fully implemented, Compute the estimation error, and provide new estimates for the whole project (**demo video sent in canvas**).

4-Provide a finalized SPMP that fits the 3 months timing constraints:

1. Introduction

In recent years, it has become clear that many students in the secondary stage face difficulties during tests and while training for final exams. This insight led to the creation of **Eduzest**, an educational application designed to provide an environment that simulates the pressures of real exams, such as final exams. With continuous advancements in artificial intelligence, we have integrated AI into our application to act as an accurate and efficient exam grader. This approach allows for faster grading with high accuracy, delivering model answers to students promptly.

Moreover, recognizing the importance of communication between students and teachers, we have added a communication page within the application. The system also compares students' grades with their classmates and sends alerts when a student's grade falls below a certain threshold. Additionally, a ranking page has been included to highlight the top-performing students.

1.1 Project Overview

This document is intended for the project members and outlines both the managerial and technical aspects of the **Eduzest** project. It serves as a planning and scheduling tool, summarizing the expected deliverables for each team.

1.2 Project Deliverables

The following items will be produced by the **Eduzest** System:

- **Software Project Management Plan:** Defines the technical and managerial processes necessary for developing and delivering the **Eduzest** system (this document).
- **Agreement between Doctor and Developers:** Represents a contract detailing the deliverables agreed upon between the doctor and the developers.
- **Requirements Analysis Document:** Describes the functional and global requirements of the system, including four models: the use case model, object model, functional model, and dynamic model.
- **Source Code:** The complete source code for all subsystems of the **Eduzest** System.

2.1 Process Model

The project is initiated on **October 20, 2024**. Major milestones include the **Doctor Project Review** on **November 29, 2024** and the **Doctor Acceptance Test** on **December 22, 2024**.

The project employs an **object-oriented design methodology** and follows the software lifecycle process, using **UML** for development. The development process is organized into several activities, with team members working together. After each activity, including testing, each team submits documents to describe their progress. These approved documents, which are considered work products, form part of the software documentation.

2.1.1 Project Planning

Project planning involves defining tasks, activities, and functions, identifying dependencies, and estimating resource requirements. The output of this phase is the **Software Project Management Plan** for the **Eduzest** System. Additionally, the project agreement is finalized after the design phase.

2.1.2 Requirements Analysis

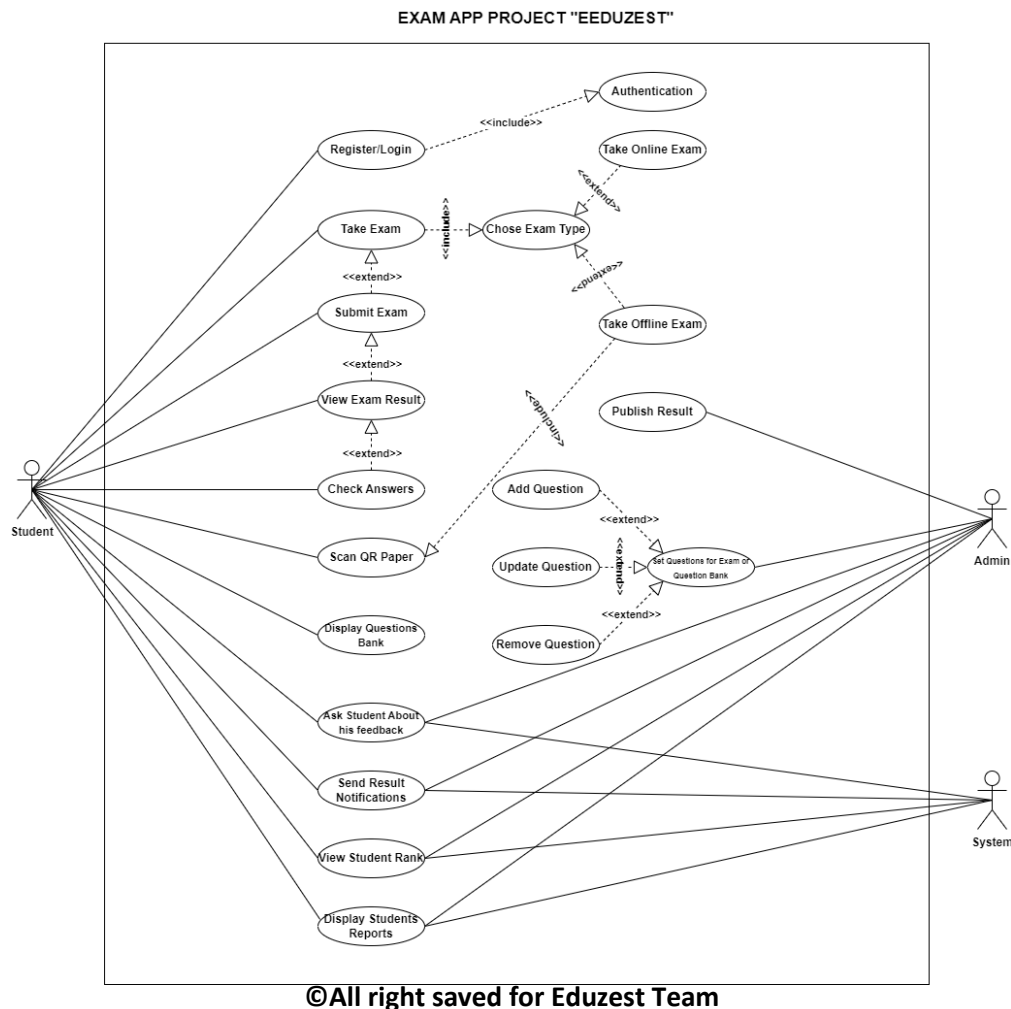
During the Requirements Analysis phase, the problem statement is reviewed for consistency, completeness, and feasibility. The process involves creating a set of models through client interactions, resulting in the requirements model. The key models include the use case model, object model, functional model, and dynamic model.

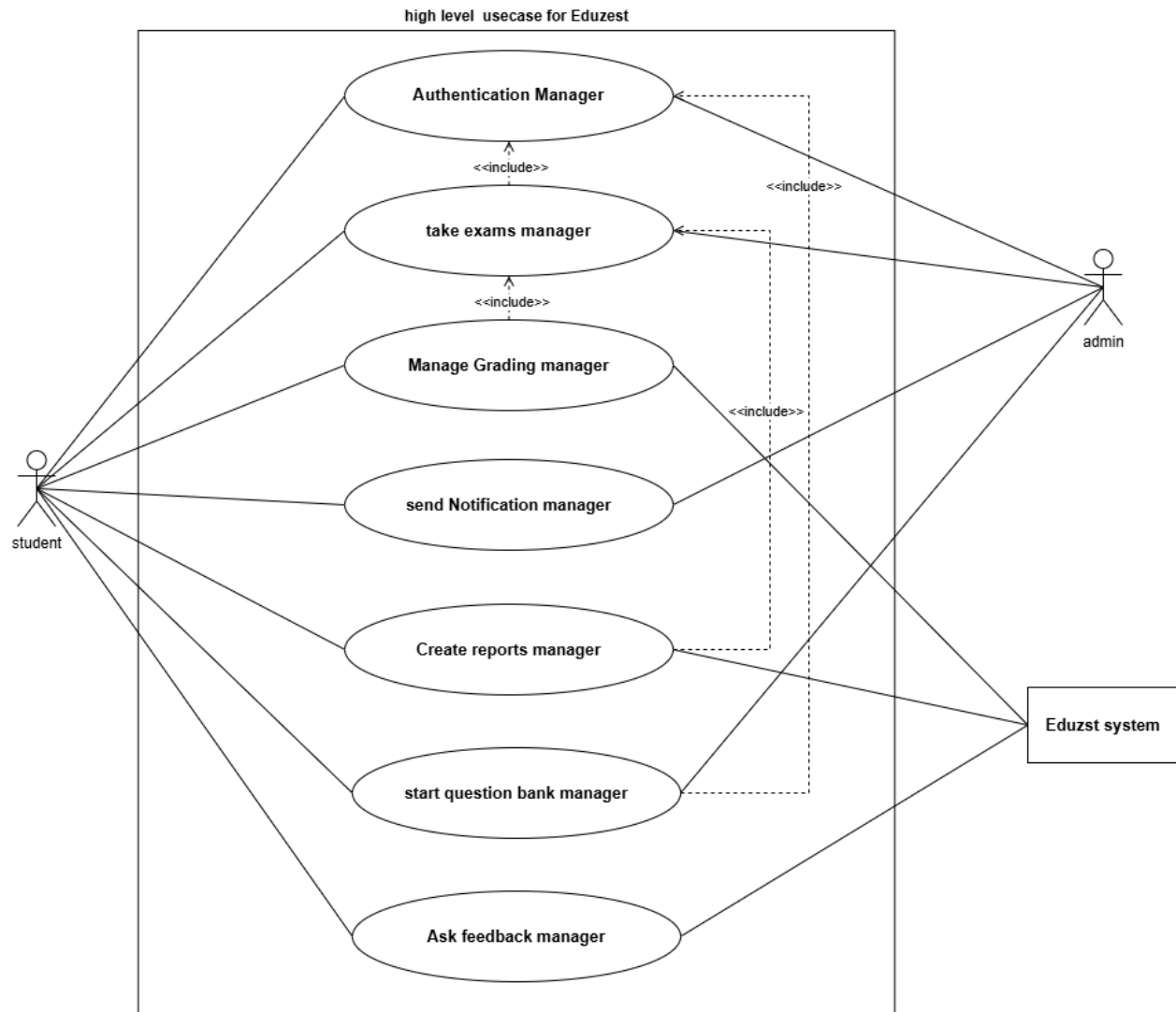
2.1.3 Analysis Review

An **Analysis Review** meeting will take place on **October 16, 2024**, from **9:00-10:20 AM**. The review consists of presentations from the project members, covering the software project management plan, requirements analysis, and design.

2.1.4 Functional Prototype Demonstration

This phase involves successfully executing a **functional prototype** of the **Eduzest** system using stubs. The functional prototype will be presented during the **internal review**.

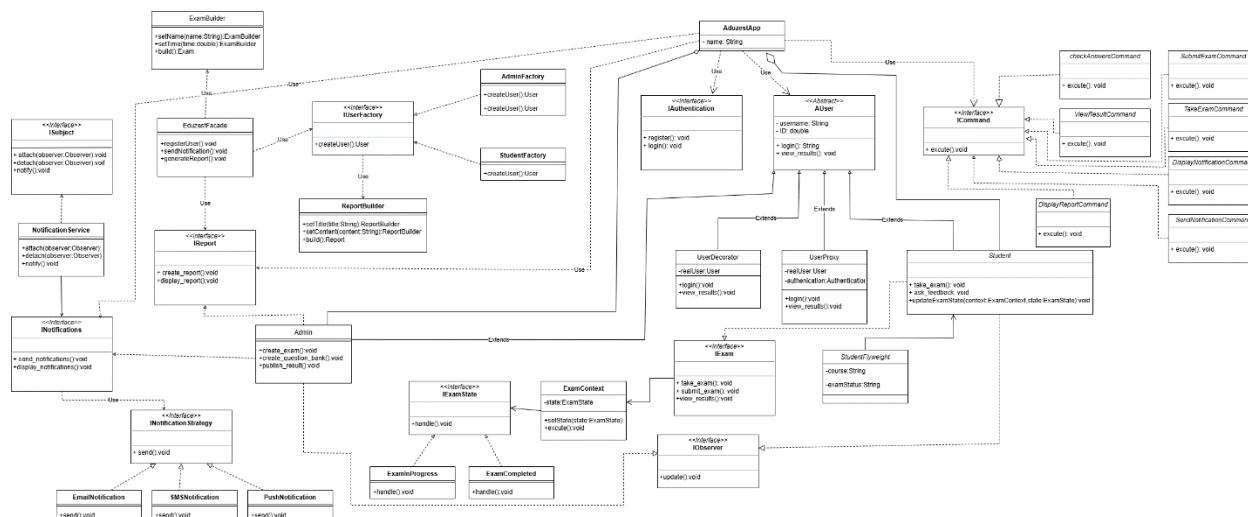




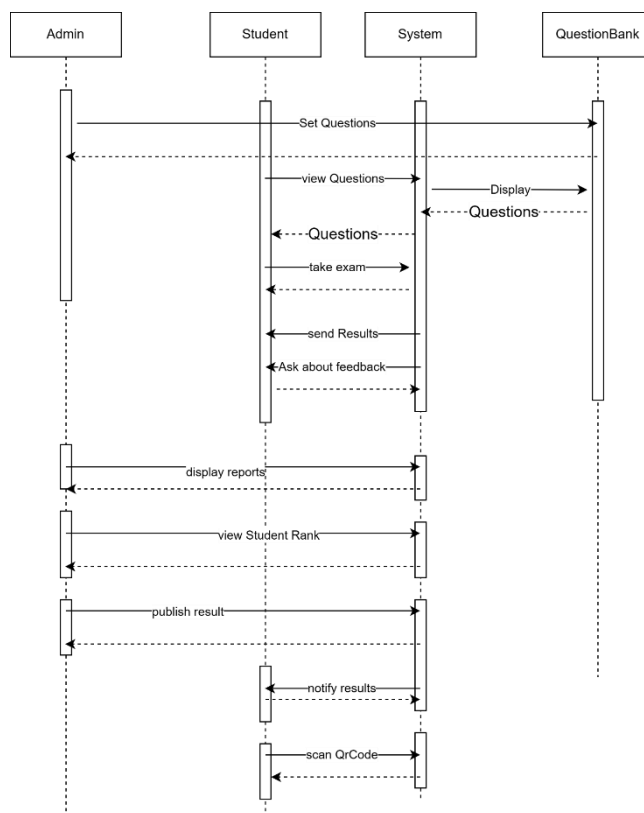
This Diagram Created By **Draw.io** for best view please follow link:

<https://shorturl.at/Yv5kS>

In this phase, new classes are added to the analysis object model, if necessary, according to the system architecture. The attributes and methods for each object are fully defined and typed.



This phase focuses on modeling the system's behavior over time, showing how it responds to events and interactions.



Features Estimation:

- **Feature 1 (Authentication Subsystem):** 12 points
- **Feature 2 (Exam Manager Subsystem):** 15 points
- **Feature 3 (Grading Manager Subsystem):** 6 points
- **Feature 4 (Report Manager):** 6 points
- **Feature 5 (Ranking Manager):** 3 points

Total Points: 42 points

Total Working Days: 84 days

3.1.1 Constraints – Deliver **Eduzest** in 3 months (from PM2 new version in next pages)

Feature	Tasks	Members	Points	Start Date	Due Date	Actual
Authentication	Login (Admin), Login (User), Register (Admin), Register (User)	Ahmed, Merna	12	26/10/2024	2/11/2024	
Exam Manager	CRUD Exam (Admin/Instructor), Submit Exam (User)	Mariam, Nadira, Mohamed	15	3/11/2024	22/11/2024	
Grading Manager	Publish Results (Instructor/Admin), Display Results (Student)	Ahmed, Mohamed	6	23/11/2024	3/12/2024	
Ranking Manager	Display Student Rank (Student)	Nadira	3	10/12/2024	15/12/2024	
Report Manager	Create Report (Admin), Display Report (Student)	Mariam, Ahmed	6	16/12/2024	27/12/2024	

Next Pages shows PM3 Updates

1. List of the adopted design patterns and their corresponding functional requirements.

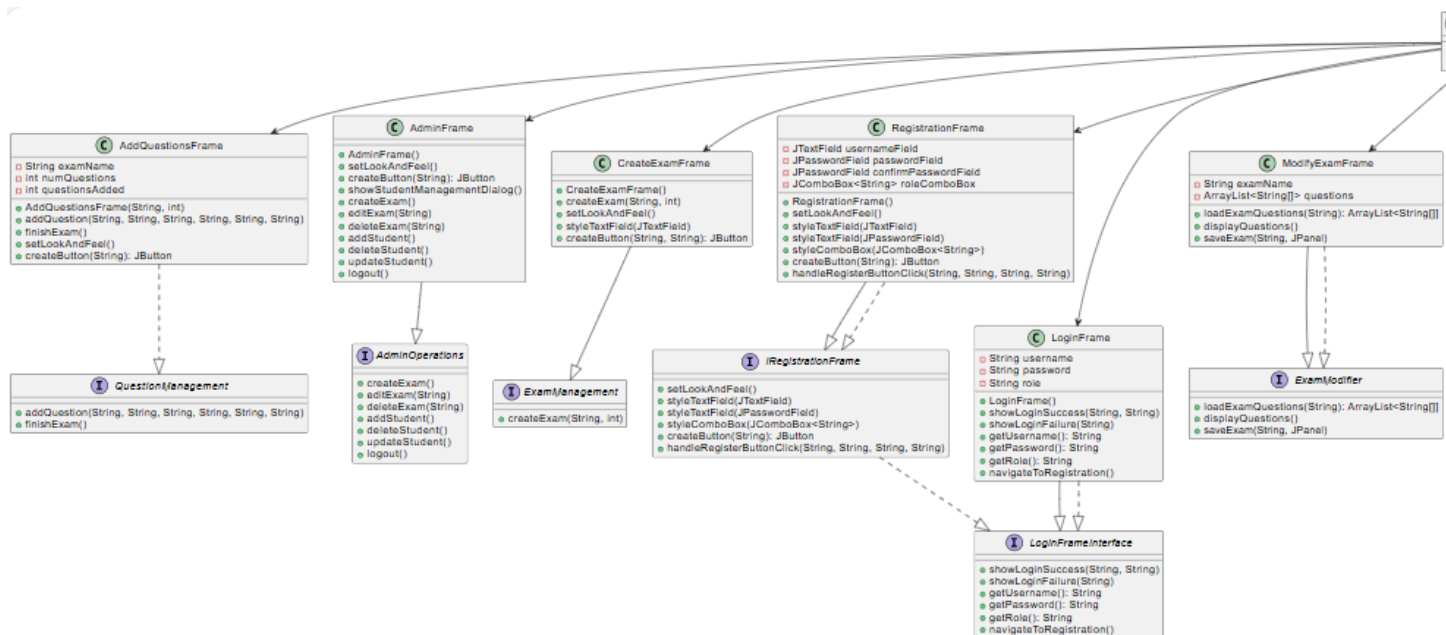
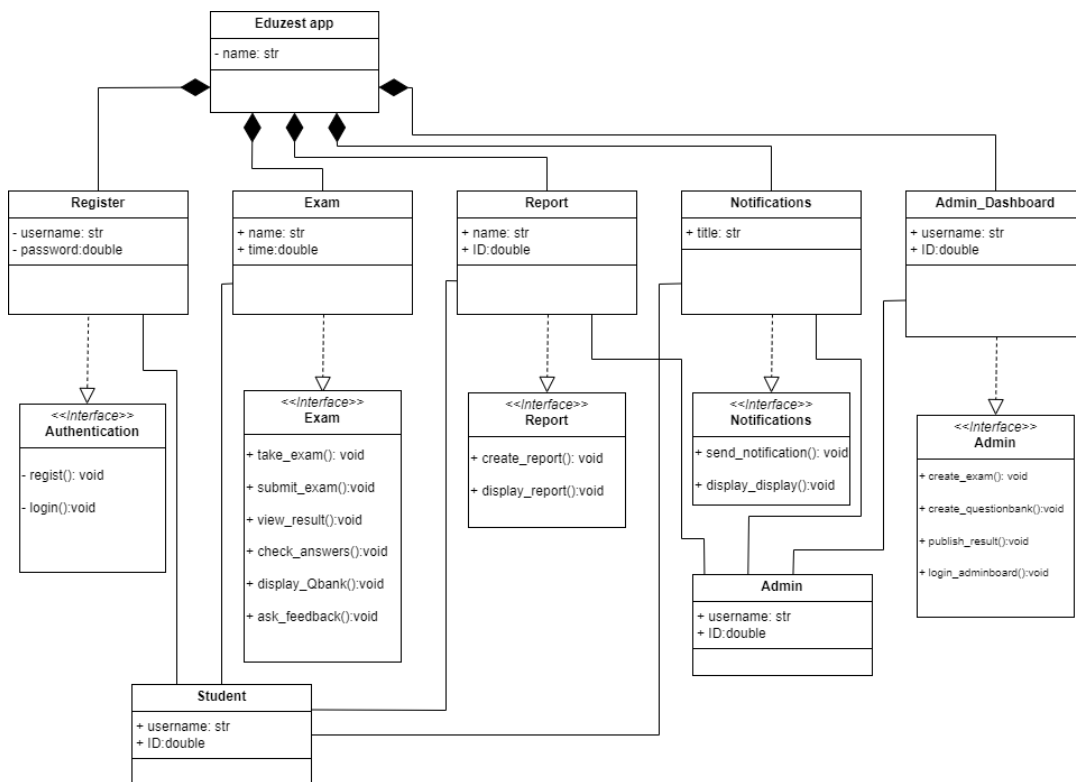
Design Pattern	Description
Abstract Factory	Allows creation of User objects (Admin and Student) through factories, ensuring correct object creation.
Proxy	Controls access to User (authentication and access control).
Command	Encapsulates requests for actions (TakeExamCommand, SubmitExamCommand).
Builder	Provides a flexible way to construct complex objects like Exam and Report step-by-step.
Strategy	Allows different notification strategies (Email, SMS, Push notifications).(feature)
Observer	Enables automatic updates (notifications) when changes occur in the system.
State	Manages the state of the Exam, allowing transitions between different exam states (InProgress, Completed).
Facade	Simplifies the interaction with complex subsystems like UserFactory, Notifications, Reports, and Exams.
Decorator	Provides dynamic behavior to User objects (logging or security-related functionality).
Chain of Responsibility	Handles authentication in a chain, with each handler processing a different authentication step (username/password, role validation).
Flyweight	Optimizes memory usage by sharing common parts of the Student object, such as course and exam status, across different instances.

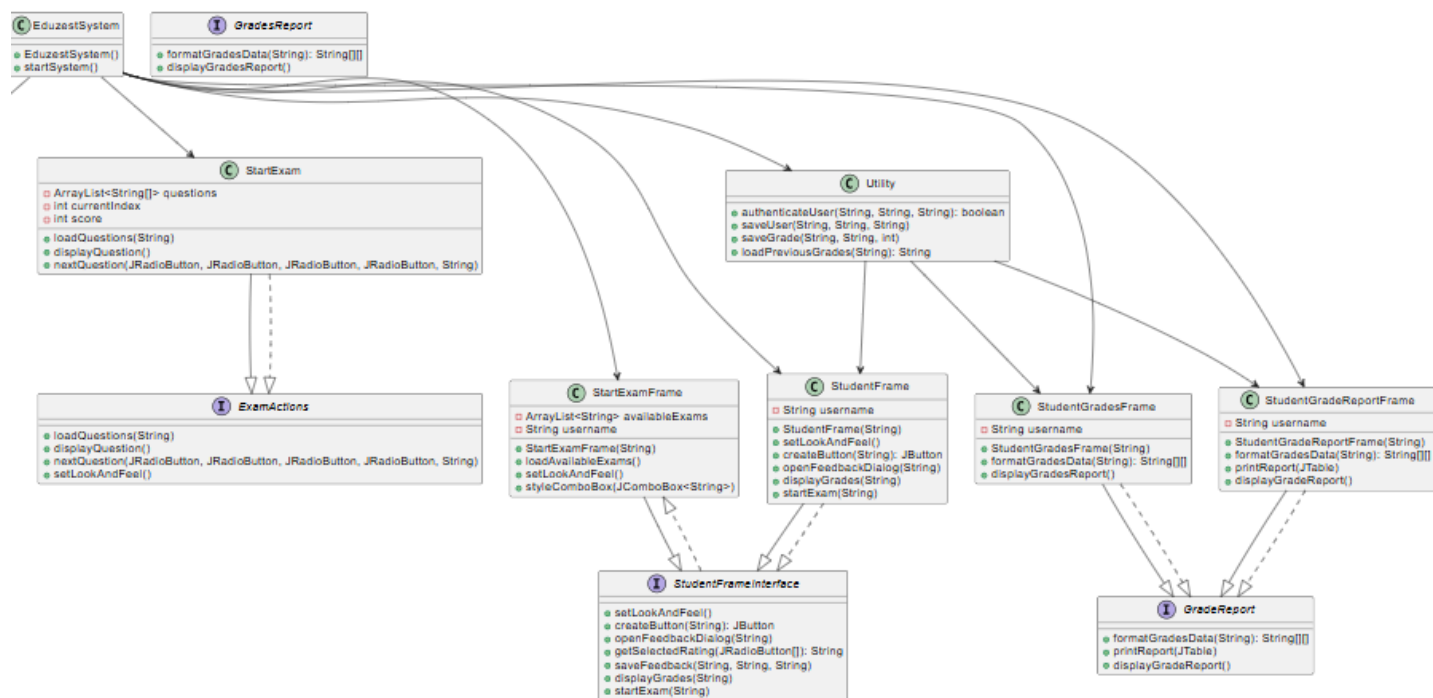
The above design patterns are applied in the provided class diagram as follows:

- 1. Abstract Factory:** **UserFactory** creates Admin and Student objects through the corresponding factory classes (**AdminFactory**, **StudentFactory**).
- 2. Proxy:** **UserProxy** class controls access to User objects and manages authentication.
- 3. Command:** The actions related to exams, reports, and notifications are encapsulated in commands such as **TakeExamCommand**, **SubmitExamCommand**, etc.
- 4. Builder:** **ExamBuilder** and **ReportBuilder** are used to construct complex objects like Exam and Report in a step-by-step manner.
- 5. Strategy:** **NotificationStrategy** interface is used to define different notification strategies like Email, SMS, and Push notifications.
- 6. Observer:** **NotificationService** notifies Admin and Student (who are Observers) when changes occur.
- 7. State:** **ExamContext** manages the state of the exam (**ExamInProgress**, **ExamCompleted**).
- 8. Facade:** **EduzestFacade** simplifies interactions with subsystems like UserFactory, Notifications, Reports, and ExamBuilder.
- 9. Decorator:** **UserDecorator** adds dynamic behavior (logging or security) to User objects.
- 10. Chain of Responsibility:** Authentication is handled in a chain, with handlers for username/password validation and role validation.
- 11. Flyweight:** The **StudentFlyweight** pattern optimizes memory usage by sharing common parts of a Student object.

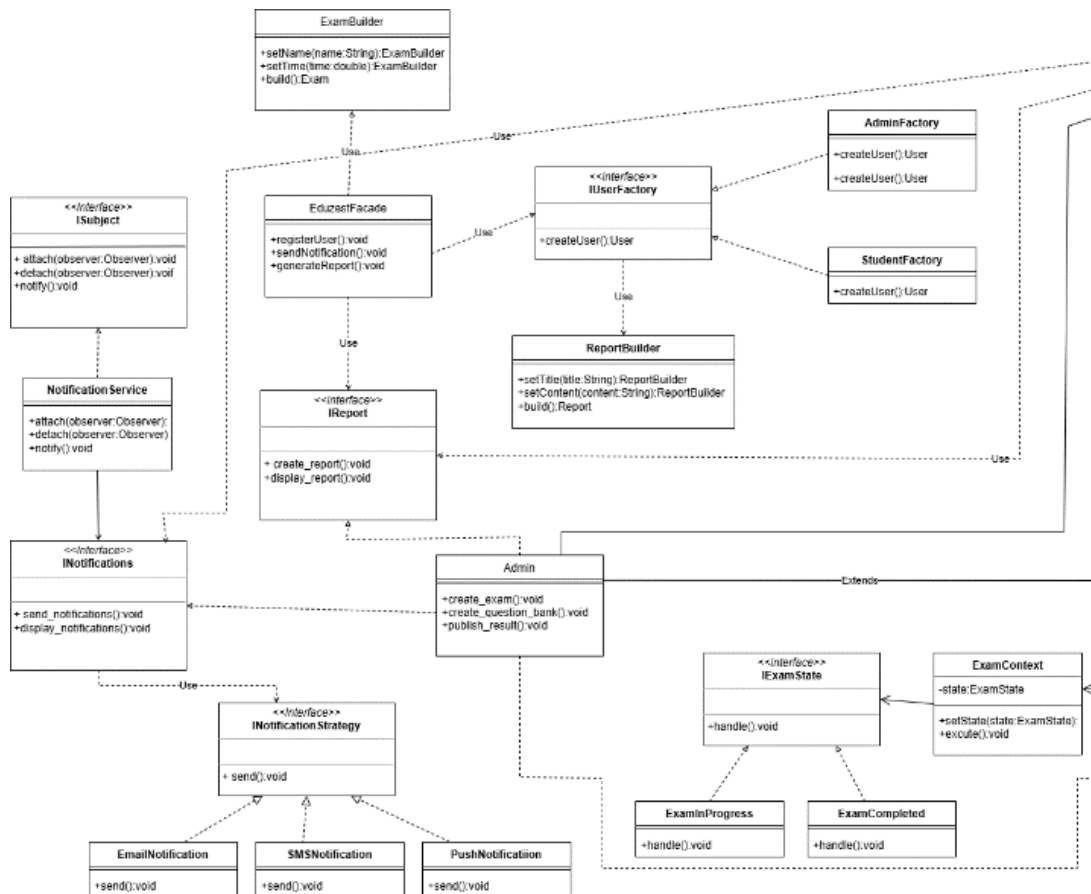
2. The class diagrams before and after including design patterns and refactoring:

2.1.The class diagrams **before** including design patterns and refactoring:





2.2. The class diagrams **after** including design patterns and refactoring:





3. Compute the Estimation Error

The project estimates were accurate overall, with all tasks being completed as planned. However, there was a small difference in the final task, which required one extra day to complete.

Error (%) = (Actual Effort-Planned Effort/Planned Effort)*100

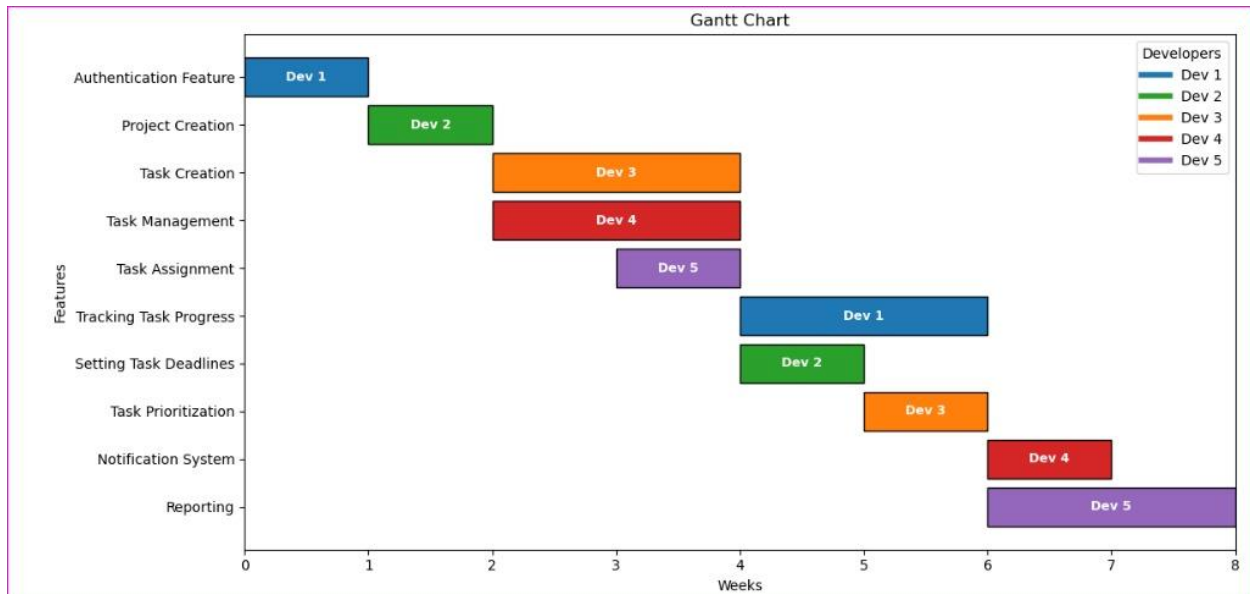
(7-6/6)*100 =16.67%

3.2 Effect on Effort

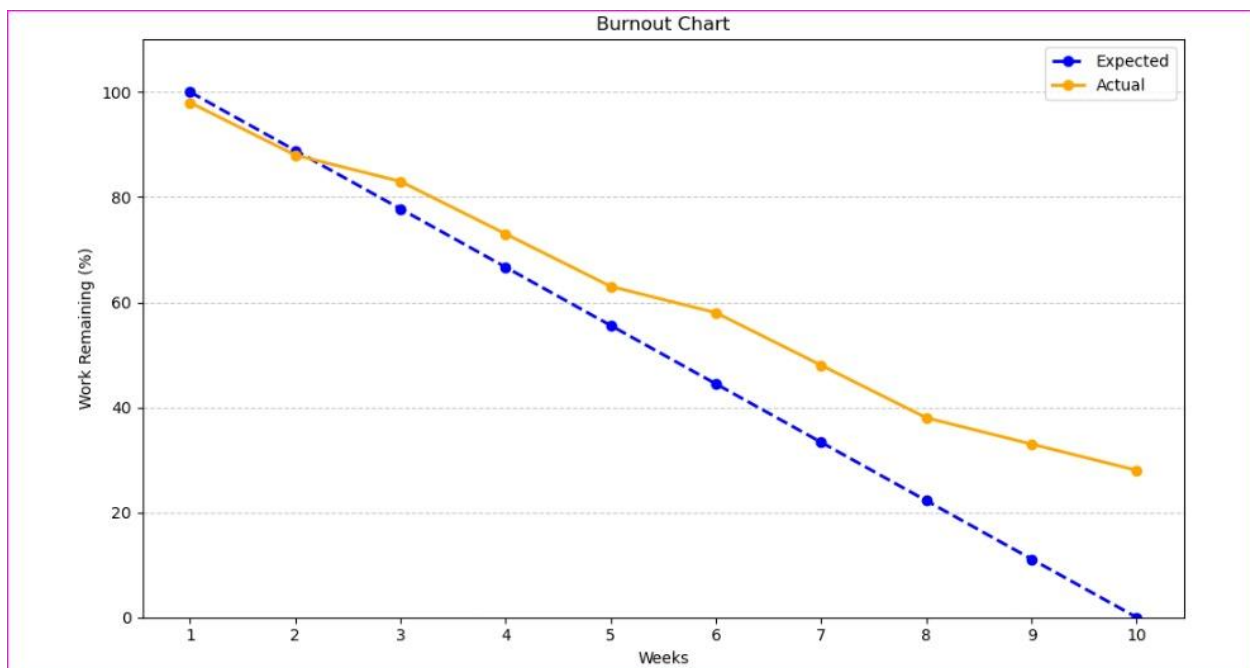
Increasing the number of working days by one.

Since this is the final task, it does not affect the previous tasks.

Gantt chart:



Burnout chart:



Features	Members	Point	Productivity	effort percentage	Error	Actual Time
Authentication	Ahmed, Merna	12	Ahmed=4point [register(User) ,Register(Admin)] Merna=8 point[Register (Admin), Login(User) , Login(Admin)]	Ahmed= 33% Merna= 66%	5%	5/11/2024
Exam Manager	Mariam, Nadira, Mohamed	15	Mariam=3 point [Create-Exam] Nadira=5 point [Read Exam] Mohamed=7 point [Delete-Exam, Update Exam , Submit Exam]	Mariam= 20% Nadira= 33% Mohamed=4 6%	3%	20/11/2024
Grading Manager	Ahmed, Mohamed	6	Ahmed=3 point [Publish Results (Instructor/Admin Mohamed=3 point [Display Results]	Ahmed= 50% Mohamed=5 0%	3%	4/12/2024
Ranking Manager	Nadira	3	Nadira=3 point [DisplayStudentRank]	Nadira= 100%	3%	13/12/2024
Report Manager	Mariam, Ahmed	6	Mariam=5 point [Create Report] Ahmed=1 point [Display Report]	Mariam= 83% Ahmed= 16%	3%	25/12/2024

Thanks 😊