



YOUR FREEDOM IN LEARNING

Transcript Manager

COMP302 - Software Engineering

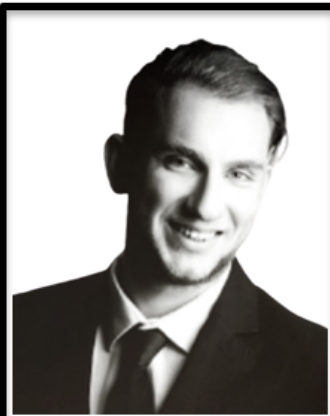
21 May 2023 – Group 2



Emir Çetin Memiş
041901027



Tuana Selen Özhazday
041901024



Emircan Yaprak
041901009

Executive Summary

The Transcript Manager project endeavors to establish an all-encompassing solution for managing academic transcripts at MEF University, targeting students who wish to track and plan their academic journey with ease. This application provides an array of functions including online execution with MEF credentials, offline execution with a MEF transcript file, course grade management, GPA calculations, course addition and subtraction simulations, and scholarship eligibility calculations. Furthermore, the software promotes achievement analysis, and visualization of academic progress, and even supports multi-language use. By synthesizing these features into a singular, user-friendly platform, the Transcript Manager ultimately aims to simplify the academic planning process, foster efficient record management, and enhance the overall student experience at MEF University.

Content

Executive Summary.....	1
Content.....	2
1. Introduction.....	3
1.1. Goals and Objectives.....	3
1.2. Statement of Scope	3
1.3. Software Context.....	4
2. Usage Scenarios.....	4
2.1. User Profiles.....	4
2.2. User Stories and Acceptance Tests.....	5
2.3. Use Case Diagram	8
2.4. Use Case Descriptions.....	8
3. Agile Development & Daily Scrum Notes.....	12
3.1. Meeting Notes.....	13
4. Architecture.....	15
4.1 Logical View	16
4.1.1 Class Diagram.....	16
4.1.2 State Diagram.....	17
4.2 Process View.....	18
4.2.1 Activity Diagram.....	18
4.2.2 Sequence Diagram.....	20
4.3 Development View.....	23
4.3.1 ER Diagram.....	23
5. Restrictions, Limitations, Constraints	23
6. User Manual	24
7. Conclusion	28

1. Introduction

This software development project is designed to tackle the complexities students at MEF University face in managing their academic records. Currently, students must navigate various fragmented tools and processes to manage and analyze their academic performance. Recognizing these challenges, our team initiated the development of a comprehensive tool, the Transcript Manager, aiming to streamline academic management and offer a unified solution.

The Transcript Manager is an innovative software tool that centralizes the management of academic records. By facilitating course grade management, GPA calculations, course addition and subtraction simulations, scholarship eligibility determination, and achievement analysis, the Transcript Manager significantly enhances the academic management experience for students. Additionally, the software offers flexible online and offline execution modes as well as filtering and sorting features catering to the diverse needs of the student population.

A crucial part of the Transcript Manager is its robust MongoDB database. MongoDB is a powerful NoSQL database that allows the system to store and retrieve various academic data efficiently, including course information, and student grades. The use of MongoDB guarantees flexibility, scalability, and performance, aligning with the diverse and dynamic needs of academic record management.

1.1. Goals and Objectives

The primary goal of this project is to design and implement an efficient and user-friendly software application, the Transcript Manager, for students of MEF University. This application aims to streamline the process of managing academic records, tracking academic performance, calculating GPAs, simulating course scenarios, and evaluating scholarship eligibility. It seeks to foster greater transparency and understanding of academic progress among students, adhering to stringent data privacy measures to ensure the safety of student information.

Our objectives include the successful design and development of the Transcript Manager within the project timeline and budget. We intend to incorporate all desired features, such as grade updates, course additions or removals, achievement analysis, and multi-language support, to make the application accessible and user-friendly. Thorough testing and debugging of the software will be conducted to guarantee a smooth and reliable user experience. Finally, positive feedback from user acceptance testing is expected, indicating satisfaction with the software's functionality and user interface.

1.2. Statement of Scope

This chapter of the report explains the statement of the scope of the Transcript Manager project.

Project Name	Transcript Manager
Scope Description	The development of an all-inclusive tool, Transcript Manager, for efficient academic record management for MEF University students.
Project Deliverables	A fully functional Transcript Manager software that is user-friendly with complete documentation and user manuals.
Acceptance Criteria	Fulfillment of all defined features, adherence to performance, security, and reliability standards, and positive user acceptance testing results.
Constraints	Data privacy compliance, system compatibility, and project timeline and budget restrictions.
Assumptions	Assumed student internet access, accurate university academic data, and university support for the project.

Table 1. Statement of The Project

1.3. Software Context

The Transcript Manager utilizes the power of Python and MongoDB to create a robust and efficient platform that enhances the overall management of academic records for students at MEF University.

Both online and offline execution modes provide students with the convenience and flexibility to access and manage their academic records according to their preferences and needs. With the Online Execution mode, students can log in using their MEF University credentials. Once authenticated, the Transcript Manager securely stores the credentials within its database for future use. This eliminates the need for students to repeatedly enter their credentials and ensures a seamless user experience.

In addition, the Transcript Manager supports Offline Execution. Students have the option to import their transcript files directly into the application. This enables them to manage their academic records independently, without relying on continuous internet access. The imported data is stored locally within the application's database, ensuring data privacy and accessibility even without an internet connection.

2. Usage Scenarios

2.1. User Profiles

The Transcript Manager aims to provide a special experience for each user profile by meeting the specific needs and requirements of students:

- **Students:** Students can log in using their MEF credentials and have the ability to view, update, and analyze their course grades, GPA, and other academic information. The platform provides features such as course filtering and sorting, simulation of course addition or removal, scholarship eligibility calculation, and visualization of achievements.

2.2. User Stories and Acceptance Tests

User stories are as follows:

- As a student, I can easily manipulate various widgets and understand what each function does on the user interface so that I can efficiently navigate and use the software.
- As a student at MEF University, I can use my MEF credentials to quickly access my transcript information from any location through the Online Execution feature in the transcript management application, so that I can manage my academic records more efficiently.
- As a student, I can filter my courses by specific criteria such as course name, semester, course code, language, ECTS information, notation, or course grade, so that I can easily find the courses I am interested in and monitor my academic progress more effectively.
- As a student, I can view my courses in a way that is most useful to me by sorting them together based on specific criteria, so that I can easily track my academic progress and stay organized.
- As a student, I can easily monitor my academic progress by checking how each course grade affects my Grade Point Average (GPA) using the "Grade Updating" feature so that I can make informed decisions about future course selections and identify areas of strength and weakness in my academic performance.
- As a student, I can make informed decisions about my course load by checking what my schedule and workload would look like with or without a particular course so that I can optimize my academic performance and achieve my goals.
- As a student who wants to apply for a scholarship, I can see which scholarship I am eligible for based on my current grades so that I can have a clear idea of what scholarship I can apply for and what grades I need to achieve.
- As a student at MEF University, I can compare my academic achievements between different semesters and have a visual representation of my progress using the Achievement Analyzer feature in the transcript management application, so that I can keep track of my academic performance.

- As a student, I can access and manage my academic records in a way that meets my needs by loading PDFs, knowing that they are safely stored offline so that I can use the transcript management application with confidence and feel secure about the privacy of my academic records.
- As a student using the transcript management application, I can load and store my favorite schemes so that I can quickly access my preferred filters and sorting options, making it easier for me to manage my academic progress.
- As a student, I can use the transcript management application in my preferred language of the MEF SIS system without any language barriers, so that I can understand all the information and instructions and efficiently manage my academic records.
- As a student, I can export a professional PDF for tracking schemes and setting targets so that I can easily share my progress with my academic advisor or other relevant parties.

Acceptance Tests for the Transcript Manager Project:

1. *Online Execution Test:*

- Enter valid MEF University credentials for online execution.
- Verify that the system successfully authenticates the user and retrieves the transcript information from the university's systems.
- Ensure that the online execution mode provides seamless access to the user's academic records.

2. *Offline Execution Test:*

- Simulate offline execution by importing a transcript file.
- Verify that the imported transcript data is accurately loaded into the system.
- Ensure that all features and functionalities work correctly in the offline execution mode.

3. *Filtering and Sorting Test:*

- Apply various filters such as course name, semester, language, and grade to the course list.
- Verify that the system accurately filters the courses based on the specified criteria.
- Test the sorting functionality by sorting courses based on different criteria like semester or course grade.

4. *Course Simulation Test:*

- Add new courses and remove existing courses from the academic record.
- Verify that the system accurately calculates the impact of course additions or removals on the GPA and academic progress.

- Test the "Grade Updating" feature to ensure the GPA is recalculated correctly when course grades are updated.

5. *Scholarship Eligibility Test:*

- Input different grades and criteria to test scholarship eligibility calculations.
- Verify that the system accurately determines the scholarships for which the student is eligible based on their grades.
- Ensure that the scholarship eligibility feature provides reliable and accurate results.

6. *Achievement Analyzer Test:*

- Get the academic achievements chart for all periods using the Achievement Analyzer feature.
- Verify that the system correctly analyzes and visualizes the student's progress over time.
- Ensure that the visual representations accurately reflect the student's academic achievements.

7. *PDF Export Test:*

- Generate a PDF report of the academic record, course details, or other relevant information.
- Verify that the exported PDF contains the required information and is generated correctly.
- Ensure that the PDF export feature provides a professional and well-formatted output.

8. *Multi-language Support Test:*

- Select different language preferences, such as English or Turkish, within the scope of MEF Sis.
- Verify that all system information, instructions, and content are displayed accurately in the selected language.
- Ensure that the multi-language support feature enhances the user experience and facilitates efficient management of academic records based on the user's language preference.

As for the nonfunctional requirements for the Transcript Manager project, they are as follows:

- User credentials and data should be securely stored and protected from unauthorized access.
- The system should be able to accommodate future growth and easily expand with new features.
- The code should follow best practices, be well-documented, and allow for modular development and updates.

2.3. User Case Diagram

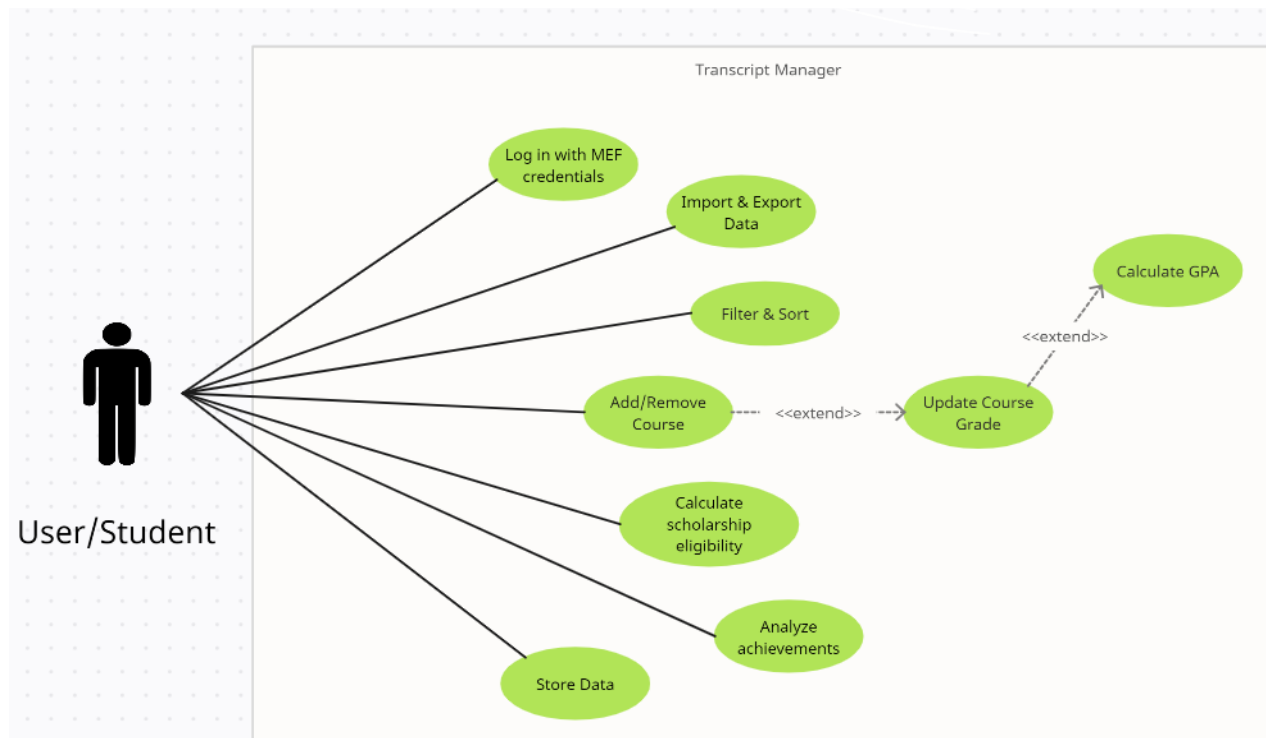


Diagram 1. The Use Case Diagram for the Project

2.4. Use Case Descriptions

Use Case	Log in with MEF credentials
Primary Actor	Student
Goal in Context	To gain secure access to academic records via MEF University credentials
Preconditions	The student has valid MEF University credentials
Scenario	1. The student enters MEF credentials 2. System verifies credentials 3. Student gains access to academic records
Exceptions	If the entered credentials are invalid, an error message is displayed

Table 2. Login Use Case Description

Use Case	Import & Export Data
Primary Actor	Student
Goal in Context	To import MEF transcript data and export academic data for tracking or sharing
Preconditions	The student has a valid MEF transcript file or has previously entered academic data
Scenario	<ol style="list-style-type: none"> 1. The student chooses to import/export data 2. System facilitates data import/export 3. Operation is completed successfully
Exceptions	If the file format is incorrect or data is not available, an error message is displayed

Table 3.Import&Export Data Use Case Description

Use Case	Filter & Sort courses
Primary Actor	Student
Goal in Context	To filter and sort courses based on specified criteria
Preconditions	The student has academic data available in the Transcript Manager
Scenario	<ol style="list-style-type: none"> 1. The student selects filtering/sorting criteria 2. The system applies the specified criteria 3. Courses are sorted/filtered accordingly
Exceptions	If no courses meet the selected criteria, a message is displayed indicating no results

Table 4.Filter&Sort Data Use Case Description

Use Case	Add & Remove course, extent Update course grade, extent Calculate GPA
Primary Actor	Student
Goal in Context	To simulate different scenarios by adding or removing courses, updating course grades, and calculating GPA accordingly
Preconditions	The student has academic data available in the Transcript Manager
Scenario	<ol style="list-style-type: none"> 1. The student adds/removes courses or updates grades 2. System recalculates GPA 3. Updated GPA is displayed
Exceptions	If the entered data is invalid or not in line with the university grading system, an error message is displayed

Table 5. Add & Remove course, extent Update course grade, extent Calculate GPA Use Case Description

Use Case	Calculate scholarship eligibility
Primary Actor	Student
Goal in Context	To calculate scholarship eligibility based on current grades and scholarship requirements and give an report
Preconditions	The student has academic data available in the Transcript Manager
Scenario	<ol style="list-style-type: none"> 1. Student requests scholarship eligibility calculation 2. System calculates eligibility based on grades and requirements 3. Eligibility status is displayed
Exceptions	If student's grades do not meet the scholarship requirements, a message is displayed indicating ineligibility

Table 6. Calculate scholarship eligibility Use Case Description

Use Case	Analyze achievements
Primary Actor	Student
Goal in Context	To visualize academic achievements progress
Preconditions	The student has academic data available in the Transcript Manager
Scenario	<ol style="list-style-type: none"> 1. Student accesses Achievement Analyzer 2. System generates comparison/visuals 3. Progress is visualized
Exceptions	If data for multiple semesters is not available, an error message is displayed

Table 7. Analyze achievements Use Case Description

Use Case	Load & Store Transcript
Primary Actor	Student
Goal in Context	To load and store transcripts with preferred filters, sorting options, and academic records for future use
Preconditions	The student has academic data available in the Transcript Manager
Scenario	<ol style="list-style-type: none"> 1. The student chooses to load/store transcript 2. System performs the operation 3. Operation is completed successfully
Exceptions	If there's an issue with storage or the specified transcript is not available, an error message is displayed

Table 8. Load & Store Transcript Use Case Description

3. Agile Scrum Development

In this part of the report, the agile development steps, daily scrum notes, and meeting minutes will be documented. The product backlog and sprint backlog can be observed in the 3 sprint cycles for the project.

Sprint 1 (5 April 2023- 18 April 2023)	Database and Login Management
Analysis & Research	Consider the "Log in with MEF credentials" use case and the tasks in the Database section
Coding	Write the codes for Database and GUI-Login
Testing	Implement unit tests for Database and GUI-Login
Documentation	Record all the changes, decisions and instructions related to these tasks

Table 9. Plan of Sprint 1

Sprint 2 (19 April 2023 - 2 May 22023)	Transcript Management and Interface Enhancement
Analysis & Research	Consider the "Load & Store Transcript", "Add & Remove course, extent Update course grade, extent Calculate GPA", and "Filter & Sort courses" use cases, along with tasks in the Interface section and the Export section
Coding	Write the codes for these tasks
Testing	Implement unit tests for these tasks
Documentation	Update all the changes, decisions, and instructions related to these tasks

Table 10. Plan of Sprint 2

Sprint 3 (3 May 2023 - 21 May 2023)	Achievement Analysis and Scholarship Calculation and PDF Export
Analysis & Research	Consider the "Analyze achievements", "Calculate scholarship eligibility", and "Import & Export Data" use cases, and the tasks in the Utilities and Constants sections
Coding	Write the codes for these tasks
Testing	Implement unit tests for these tasks
Documentation	Finalize all documentation, update the user manual, and ensure all code comments are up-to-date

Table 11. Plan of Sprint 3

3.1. Meeting Notes

- 1st meeting conducted on 15.04.2023: We planned how to approach the Transcript Manager project, we gathered some articles, some documents about the project. We reviewed our project requirements and guidelines.
- 2nd meeting conducted on 20.04.2023: We calculated daily sprint velocity. Our product manager created our product backlog and prioritized our sprint backlog. Then today's backlog tasks were expanded by the team. For today's sprint backlog, we prioritized the "Log in with MEF credentials" and "Load & Store Transcript" use cases due to the product manager's prioritization. Then the team, with the monitoring of the scrum master, played the planning game. We decided that the login handler function, the function that checks for the user that enters the credentials, was a crucial feature. As for the transcript load and store functions, we reached a consensus that these were of high importance and required a good understanding of database operations. As for the GUI for these functions, we decided it would require some work due to the need for an intuitive and user-friendly design. Since we have set our goals and established the tasks, we started to work on our sprint backlog.
- 3rd meeting conducted on 30.04.2023: We calculated today's daily sprint velocity. Our product manager assigned us the "Add & Remove course, extent Update course grade, extent Calculate GPA", and "Filter & Sort courses" tasks. We played the planning game and decided that implementing these features and their respective frontend parts would be the main focus of the next sprint, considering their importance and complexity. We started today's daily sprint by planning the functions, designing the GUI for these features, and strategizing the database operations.

The sprint cycle conducted after the 3rd meeting is planned as follows:

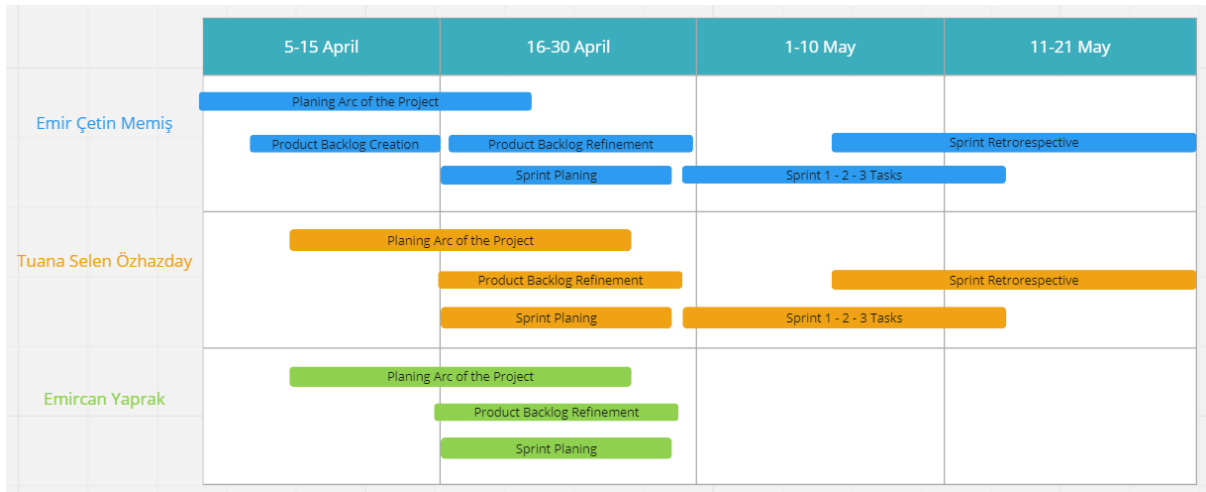


Table 11. Agile Roadmap

- 4th meeting conducted on 2.05.2023: We reviewed the completion of the first sprint, evaluated our process, and celebrated the completion of the GUI and basic database operations. However, we faced some challenges with the implementation of complex database queries. Today's sprint velocity was calculated, and the product manager assigned us tasks related to "Import & Export Data" and "Analyze achievements" use cases.
- 5th meeting conducted on 6.05.2023: We started working on the "Import & Export Data" feature. Though we faced challenges with data consistency during the import process, we implemented a validation mechanism to overcome this issue.
- 6th meeting conducted on 10.05.2023: We continued with our scheduled tasks and started developing test cases for the "Analyze achievements" feature. We also faced some challenges in visualizing the achievement data, for which we decided to use the Matplotlib library due to its wide range of options and ease of use.
- 7th meeting conducted on 15.05.2023: The "Import & Export Data" feature is almost complete and has been tested thoroughly. However, we found that the PDF exports were not formatted as expected, and we needed to spend additional time refining this functionality.
- 8th meeting conducted on 17.05.2023: We completed and thoroughly tested the "Analyze achievements" feature. We faced challenges with the performance of our data analysis algorithm, but after some optimization, we managed to improve its efficiency. We also completed the UML diagrams and included them in our project documentation.
- 9th meeting conducted on 19.05.2023: With the completion of the "Import & Export Data" feature, we started developing the "Calculate scholarship eligibility" feature. In addition, we completed a comprehensive test plan to ensure that all aspects of our software were thoroughly tested.

- 10th meeting conducted on 21.05.2023: Final sprint review and retrospective meeting were held. We successfully completed and tested the "Calculate scholarship eligibility" feature. The PDF export functionality was also finalized, and all test cases passed successfully. We revised our UML diagrams and finalized our project documentation. The challenges faced during the project were discussed, and we also explored what we learned from them and how we could use this experience in future projects.

4. Architecture

Microservices architecture has been preferred in the Transcript Manager project. This model allows our project to have a more flexible, scalable, and sustainable structure so that we can efficiently manage GUI and MongoDB integration. In this model, the application functionality is divided into independent microservices. Each microservice can manage GUI and database operations while performing a unique function.

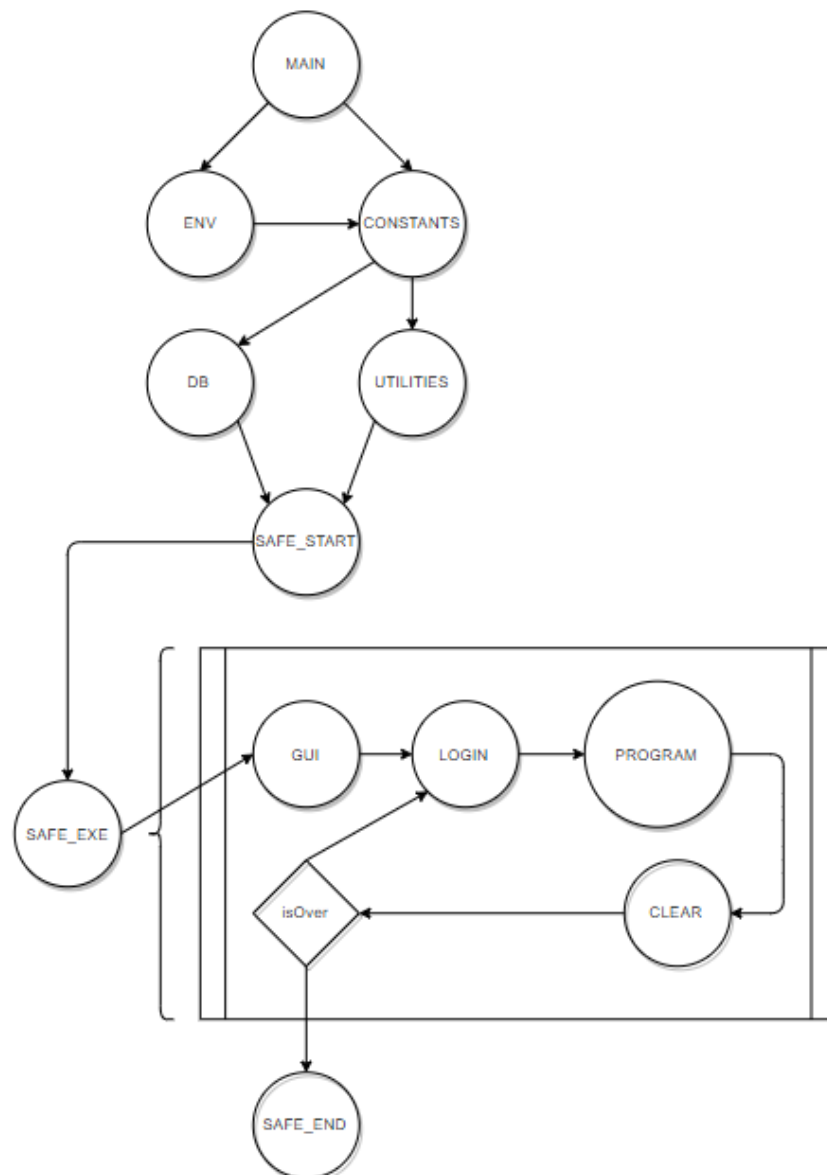


Diagram 2. The Architecture of the Project

Upon delving into the program's operational flow, the execution is initiated in the Main function, coded in Python. Upon activation, the program calls the Environment module to interpret the Constants. These Constants are then passed to the Database and Utilities modules, with the former utilizing MongoDB to manage data storage. The coalescence of the methods from these two modules triggers the `safe_start` function.

The majority of operations occur within the `safe_execute` section. Herein, the Graphical User Interface (GUI) is initialized, which facilitates user interaction with the program, and the Login process is performed. Post the login, the program embarks on its designated functionality. Simultaneously, the program performs a clean-up operation with a `clear` function to eradicate any unnecessary or residual data, ensuring the smooth running of the program. As long as `isOver` remains flagged as false, the program continues its execution.

When all tasks are deemed complete, the `safe_end` function is invoked. This function marks the termination of the process, cleanly exiting the program and ensuring all resources are appropriately released.

4.1. Logical View

This chapter of the report contains the class diagram and the state diagram for the Transcript Manager project.

4.1.1. Class Diagram

This part of the report contains the class diagram of the Transcript Manager project and the class diagram of the code written for the project.

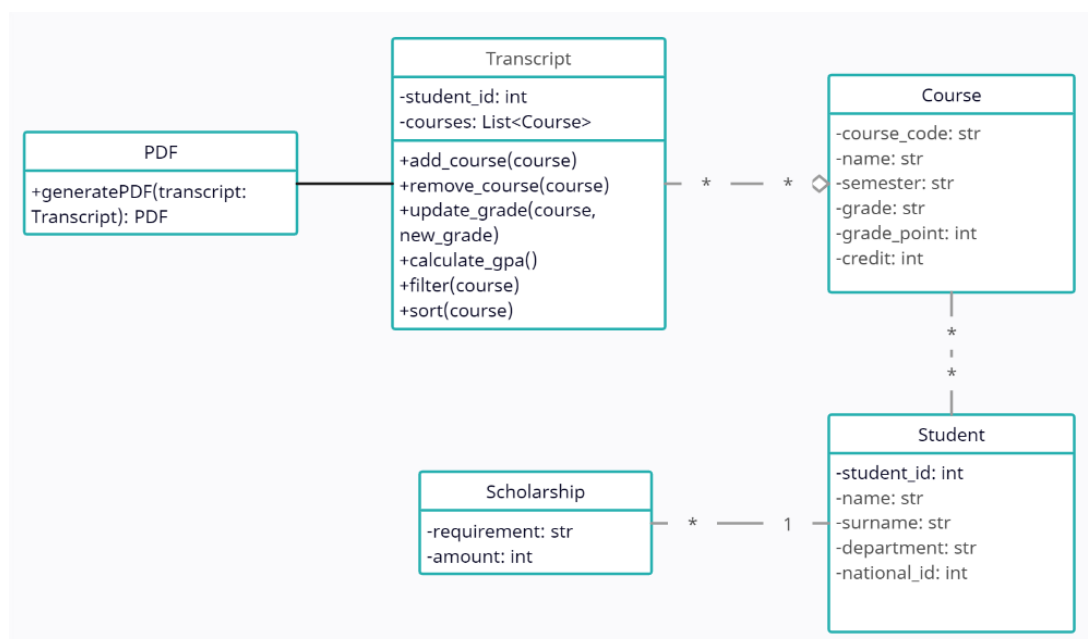


Diagram 3. Class diagram of the Transcript Manager

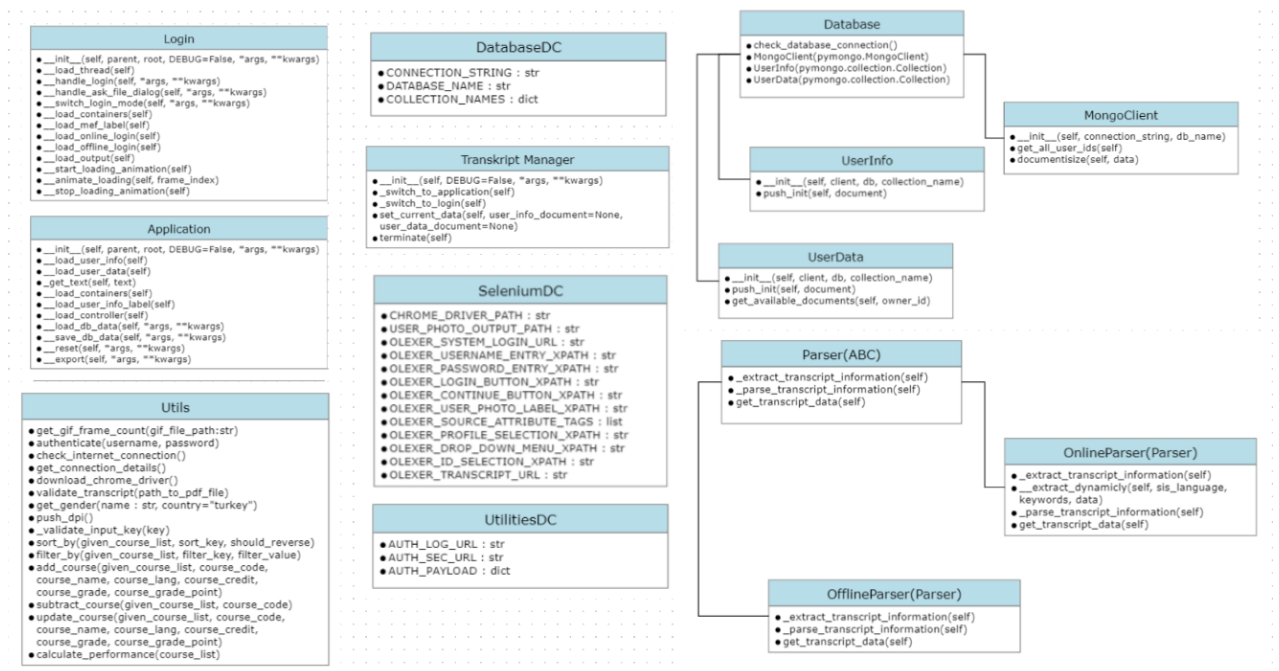


Diagram 4. Class diagram of the Transcript Manager Code

4.1.2. State Diagram

This section of the report shows the status diagram of the Transcript Manager project. The status diagram of the application is below. There are three states; Idle, Active and Performing actions. When a user opens an account, it is idle. If the user logs into their account later, their status becomes active. And if they log out of an active account, the account's status will return to the idle state.

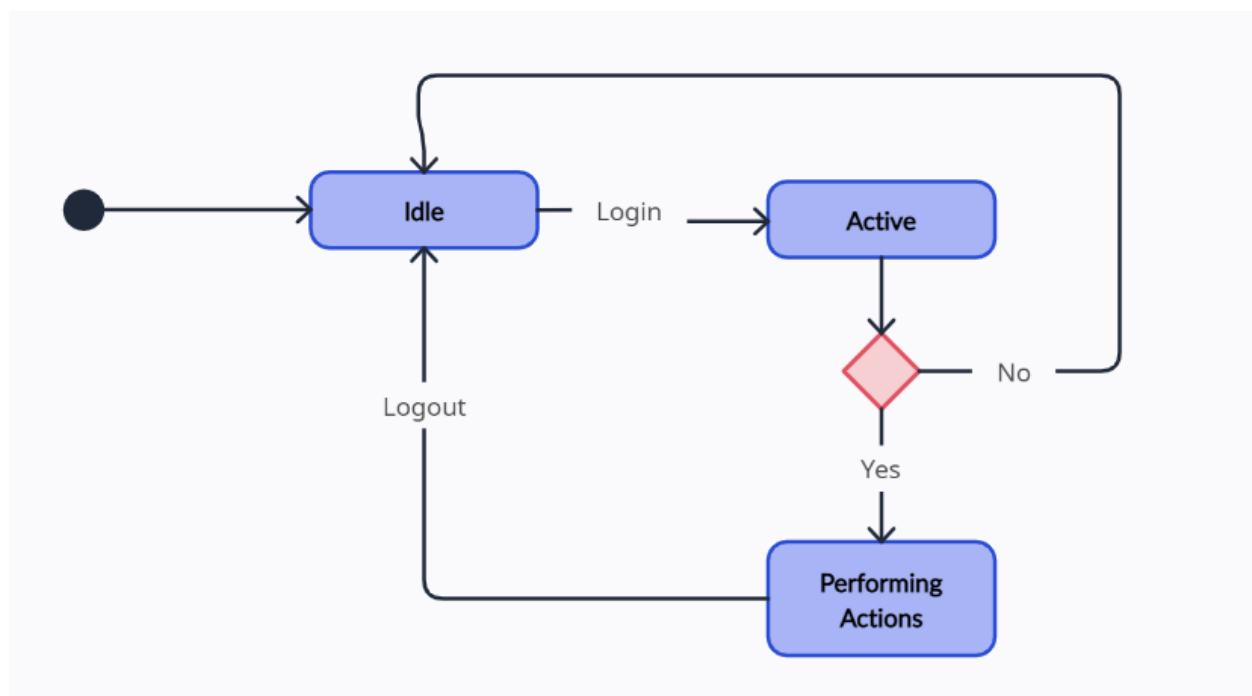


Diagram 5. The State Diagram

4.2. Process View

This chapter of the report contains the activity diagram sequence diagram and timing diagram for the Transcript Manager project.

4.2.1. Activity Diagram

This chapter of the report explains the activity diagram of the Transcript Manager project, focusing particularly on the login procedure. The process commences by verifying the existence of an active internet connection. If connectivity is confirmed, the user is encouraged to input their credentials. Upon validation of these credentials, the login process is seamlessly completed. However, if an internet connection is not detected, the user is offered an alternative method to access the system: by uploading an existing transcript file. Upon successful upload and verification of this file, the user is granted access to the system, thus concluding the login sequence.

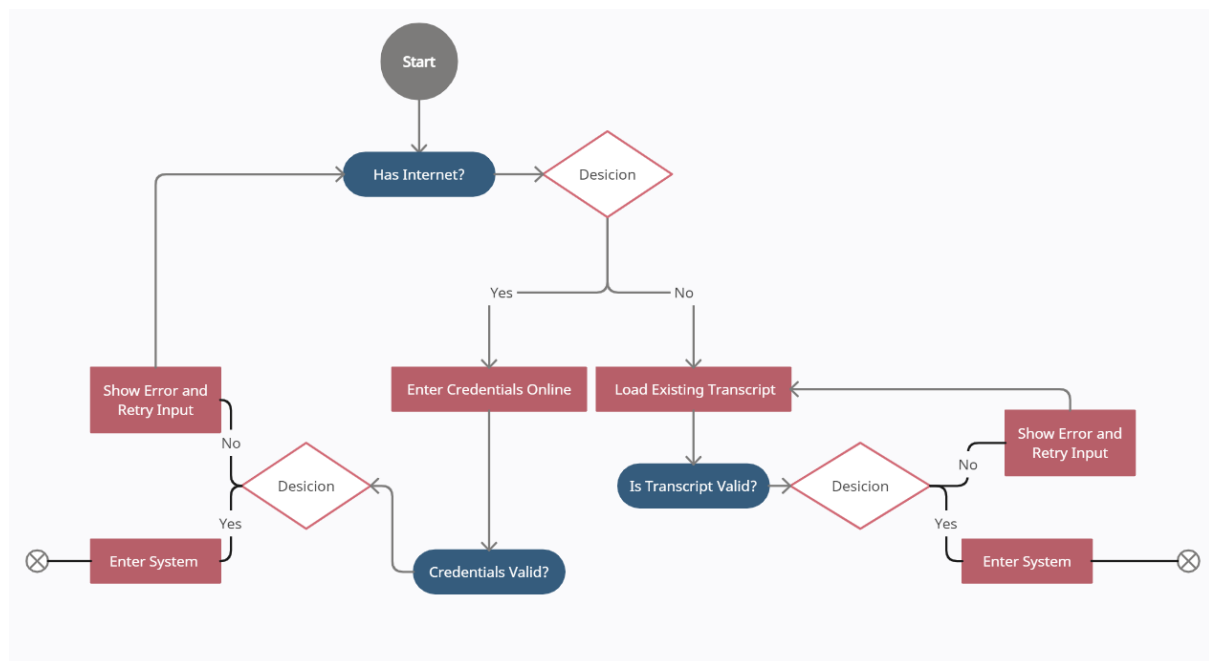


Diagram 6. The Activity Diagram for Login

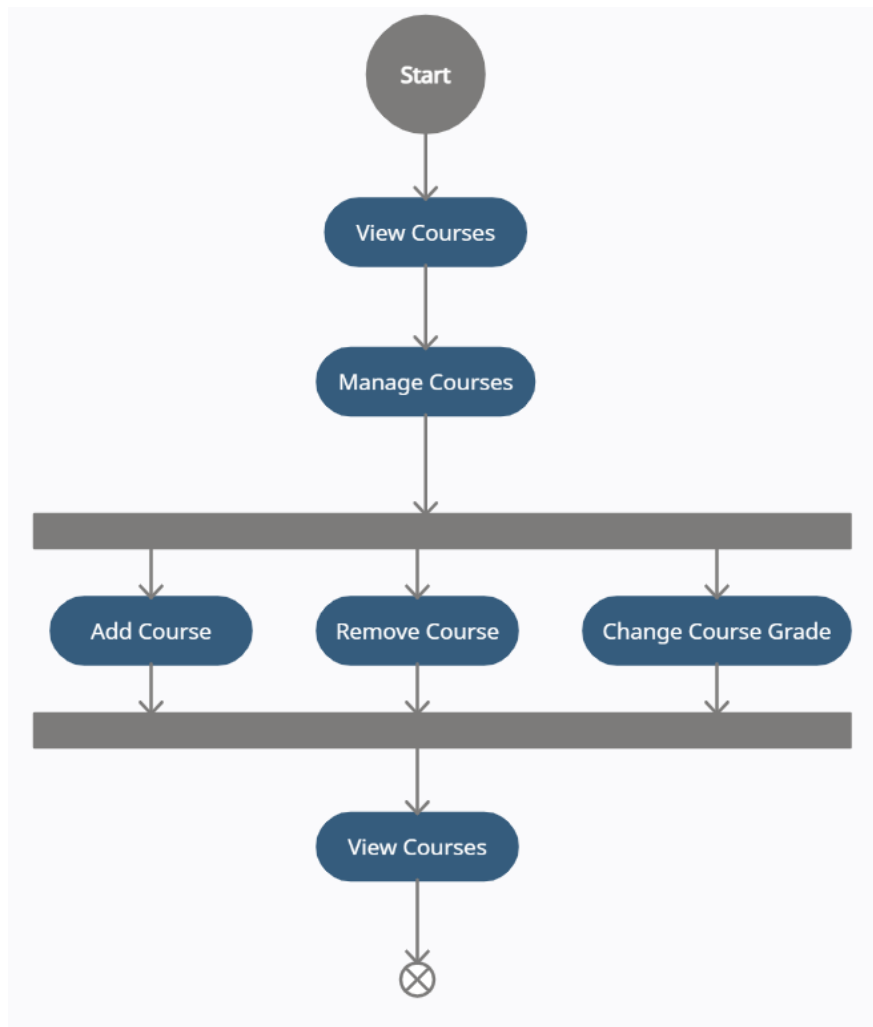


Diagram 7. The Activity Diagram for Simulating Different Course Scenarios

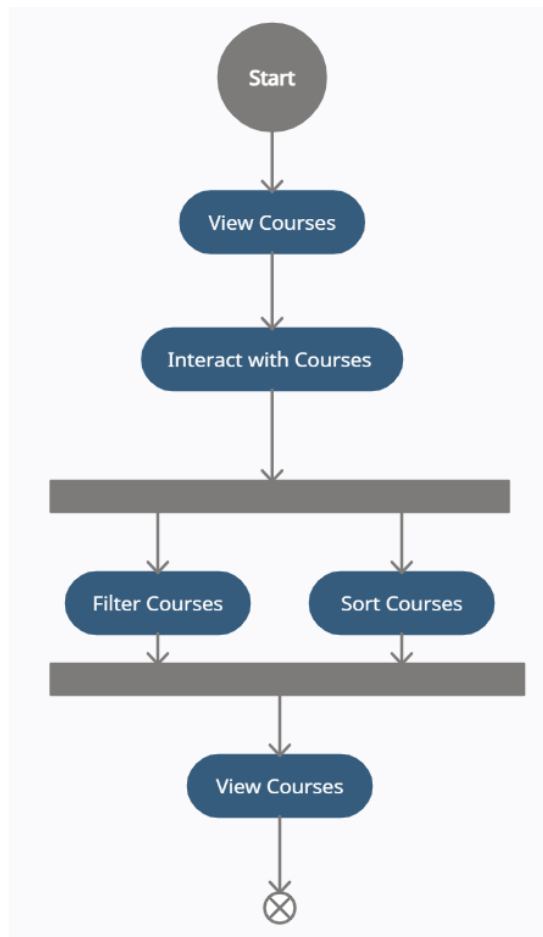


Diagram 8. The Activity Diagram for Filtering and Sorting

4.2.2. Sequence Diagram

This chapter of the report explains the sequence diagram of the Transcript Manager project.

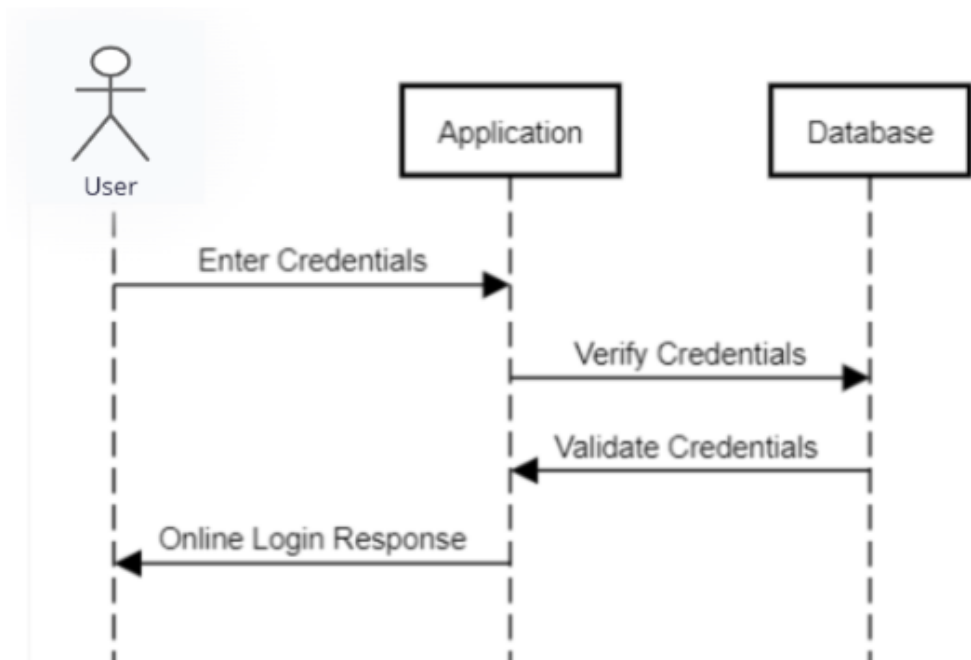


Diagram 9. The Sequence Diagram for Online Login Operation

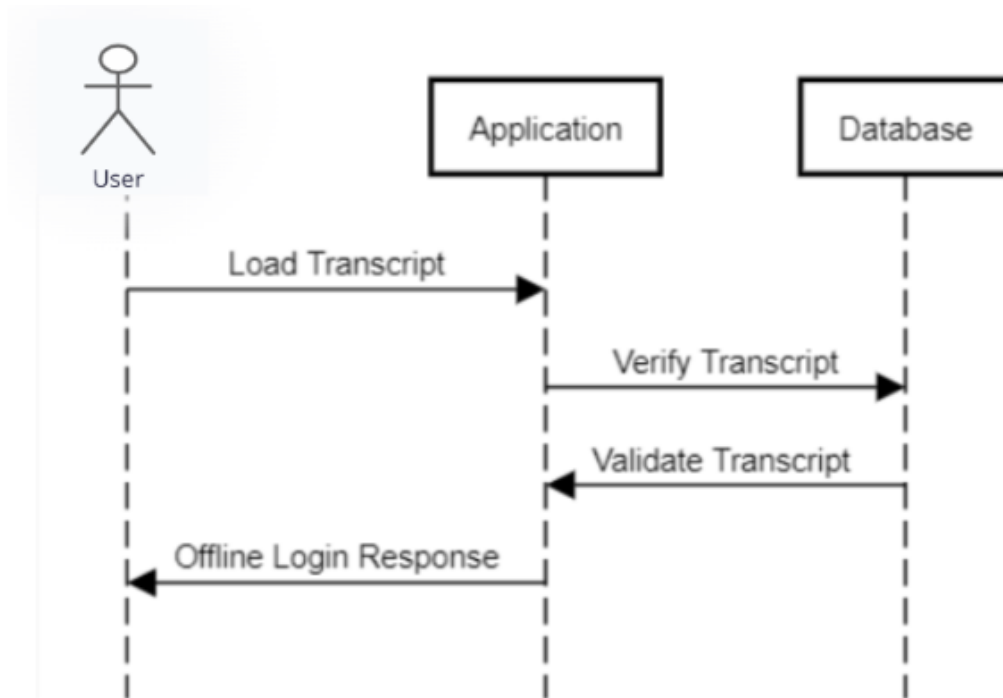


Diagram 10. The Sequence Diagram for Offline Login Operation

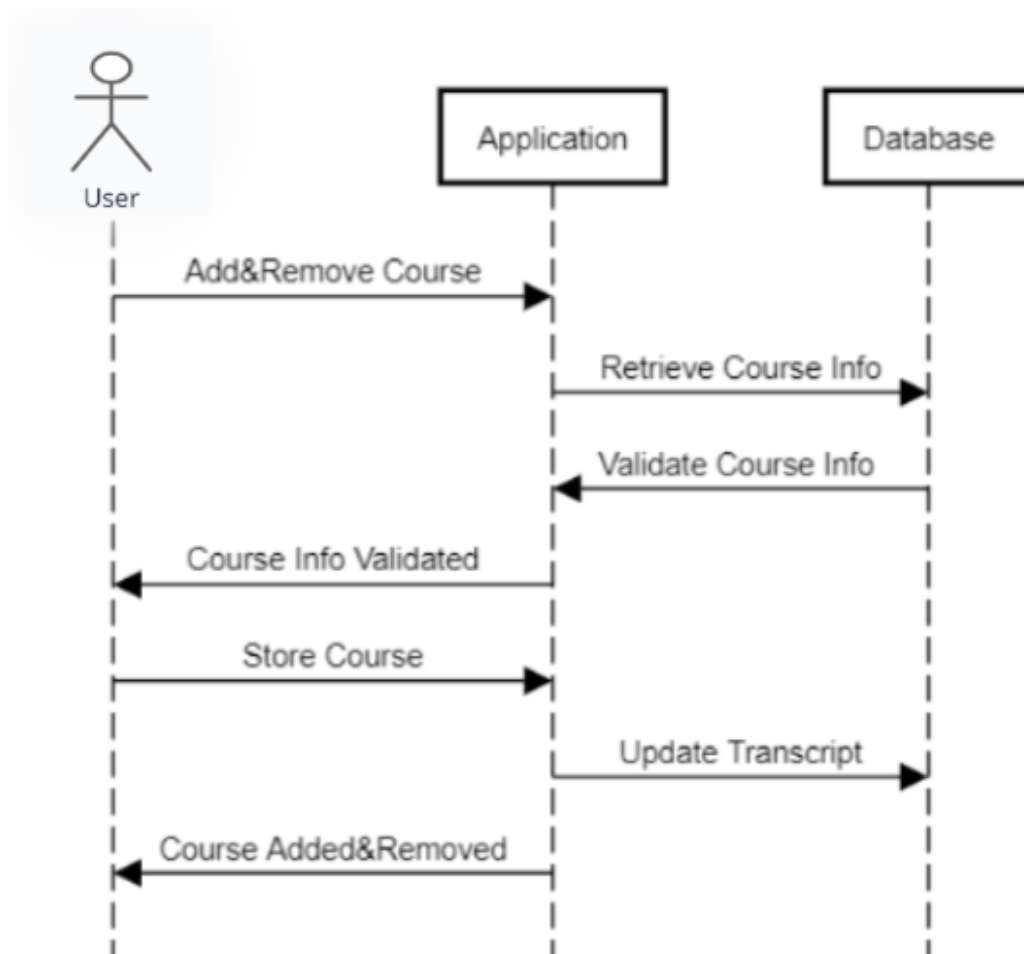


Diagram 11. The Sequence Diagram for Course Addition and Subtraction

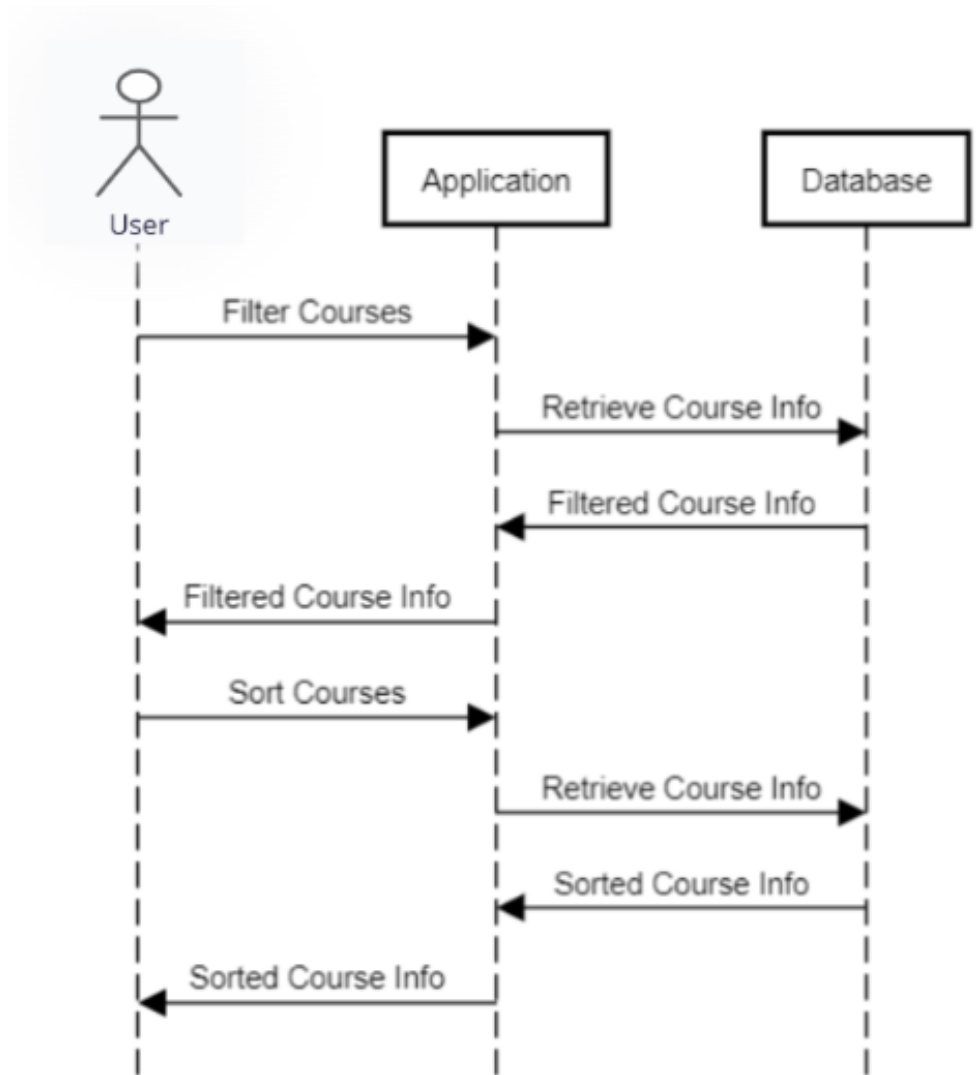


Diagram 12. The Sequence Diagram for Filtering and Sorting

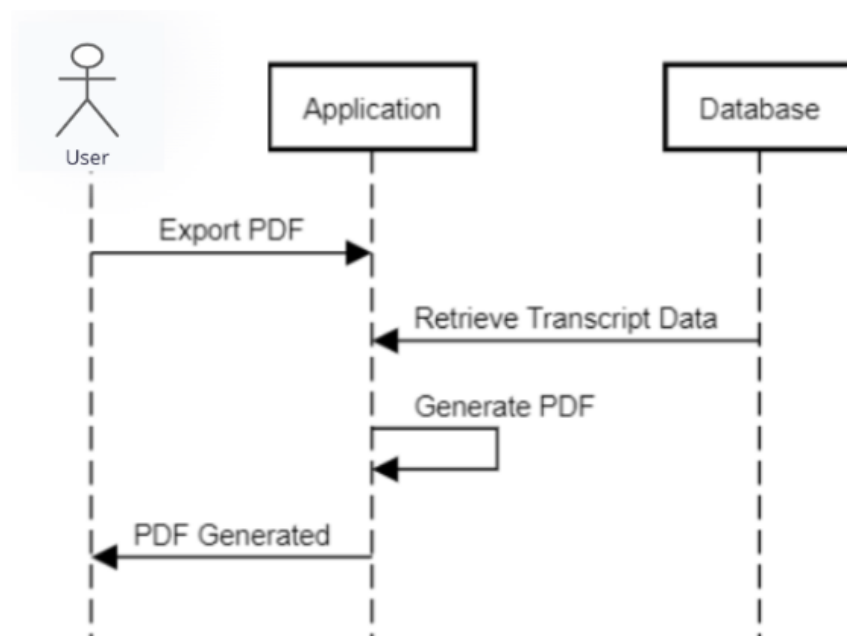


Diagram 13. The Sequence Diagram for Exporting PDF

4.3. Development View

4.3.1. ER Diagram

This chapter of the report is dedicated to the development view of the project. These will include the ER diagram.

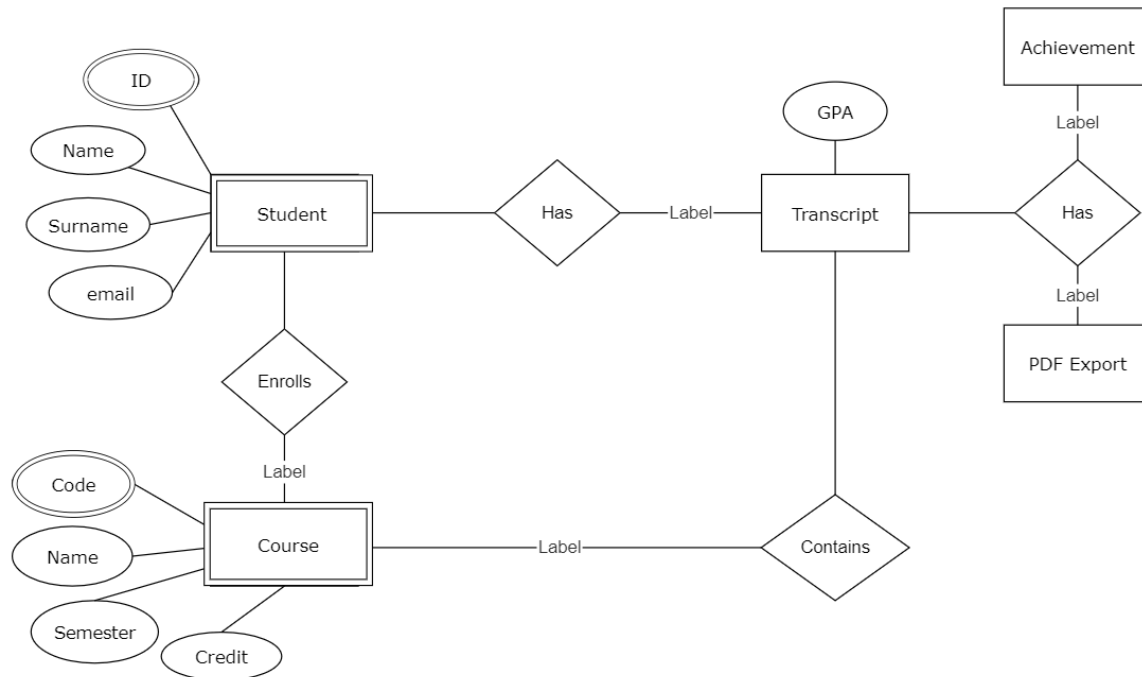


Diagram 14. The ER Diagram

5. Restrictions, Limitations, Constraints

The Transcript Manager project is subject to certain restrictions, limitations, and constraints that shape its development and implementation. These aspects include security measures, data accuracy and integration, and regulatory compliance. Security measures are of paramount importance to protect the confidentiality, integrity, and availability of academic records. The system should implement robust security protocols, including secure data transmission, encrypted storage, and strong access controls. Data accuracy and integration are critical factors for the system's functionality. The Transcript Manager relies on data from MEF University's academic records system, and thus, the integration of data should be carefully planned to ensure consistency and reliability. Regulatory compliance is another constraint that must be considered. The system should adhere to relevant laws, regulations, and privacy standards to protect user data and ensure compliance with data protection regulations.

By addressing these restrictions, limitations, and constraints, the Transcript Manager project can meet the necessary standards, provide a reliable user experience, and align with the specific requirements and regulations of MEF University.

6. User Manual

In this chapter, the user manual to operate the Transcript Manager project can be found. This chapter includes the first version of the Transcript Manager. In the demo section, the GUI will be updated.

The image shows a login screen for MEF University. It features the university's logo at the top, which consists of the letters 'MEF' in a large, blue, serif font with a red arc above them, and the word 'UNIVERSITY' in a smaller, blue, sans-serif font below. Below the logo are two input fields: 'User Name' and 'Password'. The 'User Name' field is empty, and the 'Password' field is also empty. At the bottom of the form is a green button with the word 'Login' in white text.

Figure 1. Login Screen


The image shows the same login screen as Figure 1, but with incorrect credentials entered. The 'User Name' field now contains the text 'abcabc'. The 'Password' field contains seven asterisks '*****'. Below the password field, there is a red error message that reads 'Wrong username or password !'. The green 'Login' button remains at the bottom.

Figure 2. Login Screen with wrong credentials

Faculty of Engineering / Computer Engineering				Computer Engineering		
English				Continuing		
Sort by Date	Sort by Code	Sort by Name	Sort by Language	Sort by ETCS	Sort by Notation	Sort by Grade
Reset	Suprise	Export	Restart	Exit		
0	COMP 100	Introduction to Computer Engineering	EN	3	A	12.0
1	COMP 109	Computer Programming (JAVA)	EN	6	A	24.0
2	ENG 101	English for Academic Purposes I	EN	4	B	12.0
31	MATH 115	Calculus I	EN	7	N/A	0.0
4	PHYS 103	Physics I	EN	6	A	24.0
5	PHYS 103L	Physics I Lab.	EN	2	A	8.0
6	TURK 111	Turkish Language and Literature I	TR	2	A-	7.4
7	COMP 110	Object-Oriented Programming (JAVA)	EN	6	A	24.0
8	ENG 102	English for Academic Purposes II	EN	4	A-	14.8
9	MATH 108	Discrete and Combinatorial Mathematics	EN	5	C+	11.5
Attempted Credits	Successful Credits	Included Credits	Quality Points		CGPA	
146 → 146	121 → 121	103 → 103	353.9 → 353.9		3.44 → 3.44	

Figure 3. Main Screen

Faculty of Engineering / Computer Engineering				Computer Engineering		
English				Continuing		
Sort by Date	Sort by Code	Sort by Name	Sort by Language	Sort by ETCS	Sort by Notation	Sort by Grade
Reset	Suprise	Export	Restart	Exit		
31	MATH 115	Calculus I	EN	7	N/A	0.0
10	MATH 116	Calculus II	EN	7	A	28.0
32	MATH 213	Differential Equations	EN	7	W	0.0
1	COMP 109	Computer Programming (JAVA)	EN	6	A	24.0
4	PHYS 103	Physics I	EN	6	A	24.0
7	COMP 110	Object-Oriented Programming (JAVA)	EN	6	A	24.0
11	PHYS 104	Physics II	EN	6	A	24.0
13	COMP 201	Data Structures and Algorithms	EN	6	A	24.0
14	COMP 205	Systems Programming	EN	6	A	24.0
30	EE 203	Digital Systems Design	EN	6	N/A	0.0
Attempted Credits	Successful Credits	Included Credits	Quality Points		CGPA	
146 → 146	121 → 121	103 → 103	353.9 → 364.4		3.44 → 3.54	

Figure 5. Sort by ECTS

Faculty of Engineering / Computer Engineering				Computer Engineering		
English				Continuing		
Sort by Date	Sort by Code	Sort by Name	Sort by Language	Sort by ETCS	Sort by Notation	Sort by Grade
Reset	Suprise	Export	Restart	Exit		
10	MATH 116	Calculus II	EN	7	A	28.0
1	COMP 109	Computer Programming (JAVA)	EN	6	A	24.0
4	PHYS 103	Physics I	EN	6	A	24.0
7	COMP 110	Object-Oriented Programming (JAVA)	EN	6	A	24.0
11	PHYS 104	Physics II	EN	6	A	24.0
13	COMP 201	Data Structures and Algorithms	EN	6	A	24.0
14	COMP 205	Systems Programming	EN	6	A	24.0
20	COMP 204	Programming Studio	EN	6	A	24.0
19	MATH 211	Linear Algebra	EN	6	A-	22.2
9	MATH 108	Discrete and Combinatorial Mathematics	EN	5	A	20.0
Attempted Credits		Successful Credits	Included Credits	Quality Points		CGPA
146 → 146		121 → 121	103 → 103	353.9 → 364.4		3.44 → 3.54

Figure 5. Sort by Grade

Faculty of Engineering / Computer Engineering				Computer Engineering		
English				Continuing		
Sort by Date	Sort by Code	Sort by Name	Sort by Language	Sort by ETCS	Sort by Notation	Sort by Grade
Reset	Suprise	Export	Restart	Exit		
6	TURK 111	Turkish Language and Literature I	TR	2	B-	5.4
12	PHYS 104L	Physics II Lab	EN	2	A	8.0
11	PHYS 104	Physics II	EN	6	A	24.0
5	PHYS 103L	Physics I Lab.	EN	2	A	8.0
4	PHYS 103	Physics I	EN	6	A	24.0
33	MATH 321	Automata Theory and Formal Language	EN	6	W	0.0
27	MATH 224	Probability and Statistics for Engineering	EN	6	C-	10.2
32	MATH 213	Differential Equations	EN	7	W	0.0
19	MATH 211	Linear Algebra	EN	6	A-	22.2
10	MATH 116	Calculus II	EN	7	A	28.0
Attempted Credits		Successful Credits	Included Credits	Quality Points		CGPA
146 → 146		121 → 121	103 → 103	353.9 → 364.4		3.44 → 3.54

Figure 5. Sort by Code

7. Conclusion

In conclusion, the development of the Transcript Manager project aims to provide MEF University students with an efficient and user-friendly platform for managing their academic records. Through the use of modern technologies such as Python, MongoDB, and various libraries, the project aims to streamline the process of accessing, organizing, and analyzing academic data.

The Transcript Manager offers a range of features including online and offline execution modes, filtering and sorting options, course simulation, GPA calculation, scholarship eligibility assessment, achievement analysis, PDF export, and multi-language support. These features cater to the diverse needs of students and empower them to effectively track their academic progress, make informed decisions, and share their records with advisors or other relevant parties.