CS/SE/CE 3354 Software Engineering
Final Project Deliverable 1

Interactive and Visual Academic Standing Integration

(IVASI)

Kai Chang
Kamil Huddha
Reid Jamesmeyer
Raghav Mukundan
Dhruvin Patel
Suhas Shivaraju
Ananthram Tekkalakota
Yisak Worku

Project Goal:

We will be implementing the engineering design process to create a program that can be seamlessly integrated into a classroom learning environment. In simple terms, this program will convert a course's assignments, quizzes, and other related tasks into a video game-like format. Our aim is to design a program that would help students stay engaged with the content as well as visualize their academic progress/standing in the class (i.e. strengths, weaknesses, grades, completion progress).

Motivation:

Our group decided on IVASI because we shared a desire to improve the mundane classroom environment that makes it hard for students to stay engaged. In addition, we are students of the same university as well as having a shared field of study, therefore we can all relate to having too many deadlines, learning topics, and notes clouding our headspace. Our group aims to develop a solution that eases student anxiety by providing students with a clean (well organized), visual, and interactive tool that encourages more classroom engagement and makes completing tasks enjoyable and rewarding. We expect this design to be used in an academic setting such as public and private schools, as well as higher education institutions.

Deliverable 2 (and 1) Task Delegation/Contribution:

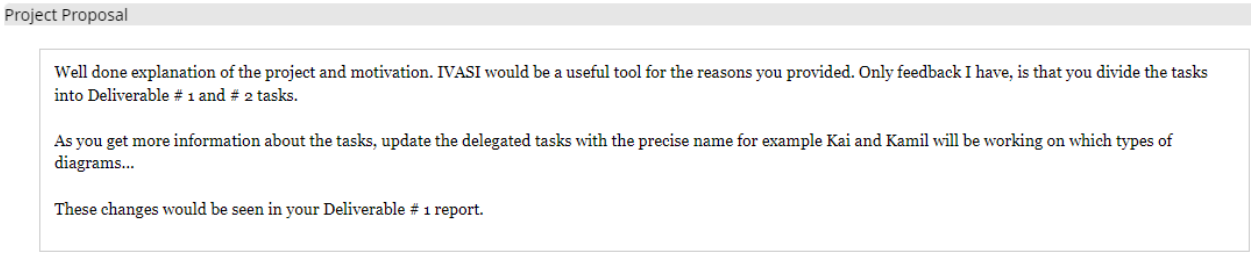
- *Reid Jamesmeyer:*
    - Deliverable #1:
        - Convert our Project Proposal into a rough draft for the Deliverable #1 Report.
        - Complete the Case Diagram portion of the Deliverable #1 Report.
        - Create the GitHub repository to be used for the project
        - Add all group members as collaborators
        - Complete the Project Scope to be uploaded to GitHub
    - Deliverable #2:
        - Convert our Deliverable #1 Report document into a rough draft for the Deliverable #2 report
        - Finalize Deliverable #2 Report
        - Provide a project schedule
        - Calculate and provide the Cost, Effort, and Pricing Estimation Report as well as:
            - Provide cost of hardware products
            - Provide cost of software products
            - Provide cost of personnel
        - Create a test plan for the calculate_student_progress method
        - Execute the test plan for calculate_student_progress method
            - Record results
        - Compare work with similar designs
        - Write the Project Summary
        - Submit Project to ELearning and GitHub
- *Kai Chang:*

    - Deliverable #1:
        - Complete the Sequence Diagram (Contact Teacher)
        - Complete the Sequence Diagram (Complete Task/Assignment)
    - Deliverable #2:
        - Convert a portion of the docs to powerpoint
        - Made slide 1 through 17
- *Kamil Hudda:*
    - Deliverable #1:
        - Complete the Class Diagram
    - Deliverable #2:
        - Corrected the errors made in the class diagram
        - Edited slides 14 and 21
- *Raghav Mukundan:*
    - Deliverable #1:
        - Complete the Sequence Diagram (View Class Completion Progress)
        - Finalized and Committed the Project Scope File to GitHub
    - Deliverable #2:
        - Convert a portion of the docs to powerpoint
        - Made slides 19-22
        - Edited slides 2,3,4,15,16,17
- *Dhruvin Patel:*
    - Deliverable #1:
        - Complete the Sequence Diagram (View Personal Info)
    - Deliverable #2:
        - Made and edited slides 17 and 18
        - Helped fix view personal info diagram
        - Added transitions and effects to slides
- *Suhas Shivaraju:*
    - Deliverable #1:
        - Complete the Sequence Diagram (View Academic Standing)
    - Deliverable #2:
        - Made slides 17,18, 21
        - Edited slides 2,3,4,5,6,7,8
- *Yisak Worku:*
    - Deliverable #1:
        - Complete the Architectural Design (Repository Architecture Pattern)
    - Deliverable #2:
        - Added visual to side 4,7
        - Edit slide 18

- *Ananthram Tekkalakota:*
    - Deliverable #1:
        - Helped make Sequence diagrams(View Personal and Academic Standing)

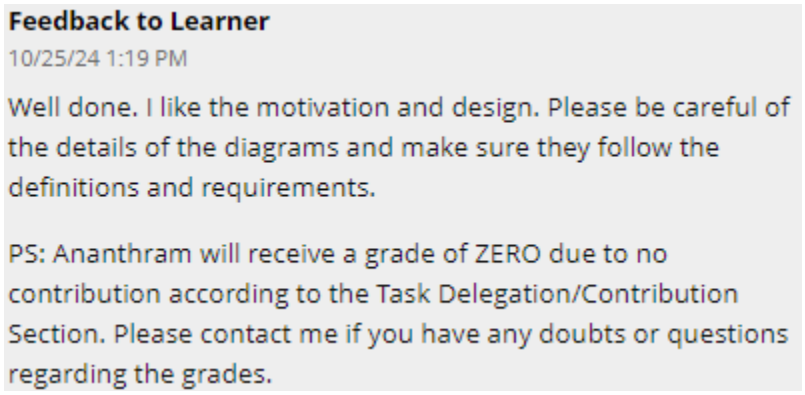
- Deliverable #2:
  - Edited slides 9,14,17,20,21
  - Made slides 18
  - Added visuals to slides:10,11
  - Fixed view personal info diagram
  - Added transitions between slides

Proposal Feedback:



Based on the feedback that received for our proposal, we have revised our task delegation documentation which now only includes tasks relevant to the Deliverable #1 Report.

Deliverable #1 Feedback:



Feedback to Learner

10/25/24 1:19 PM

Well done. I like the motivation and design. Please be careful of the details of the diagrams and make sure they follow the definitions and requirements.

PS: Ananthram will receive a grade of ZERO due to no contribution according to the Task Delegation/Contribution Section. Please contact me if you have any doubts or questions regarding the grades.

Based on the feedback we received from our Deliverable #1, we have revised and edited the details in our diagrams to better match and align with the project definitions and requirements. Following feedback suggestions we have also revised the project functional and non-functional requirements to be more specific and organized, improving readability.

GitHub Repository Link:

https://github.com/ReidUTD/3354-ProjectGroup3.git

Software Process Model:

Our group has chosen to implement the Prototyping Evolutionary Model. Our project has a clear goal of how we want our software to impact students and teachers, but we have left a lot of the actual specifications open to interpretation. We chose this model because it is the model most capable of producing an efficient and stable product even when software, environment, and efficiency variables are unknown.

Functional Requirements:

1. A user shall be able to query any/all parts of their profile whenever they need to.
2. The system shall generate a user's (student's) metadata for the user to view using graphs, progress bars, large buttons, and minimal text.
3. Each teacher/professor using the system shall be uniquely identified by a special tag or ID
4. The system shall allow any teacher user to edit or update a student's metadata.
5. The system shall allow any teacher user to create/update/remove any tasks required of the students.
6. A teacher and student user shall be able to communicate with each other via private messaging.

Non-Functional Requirements:

Product Requirements:

- Dependability Requirements: The system shall be available to any educational organization and be fully functional during work day hours which means the system shall maintain a 99% uptime to ensure that any user can access the system without any interruption.
- Performance Requirements: The system shall generate user metadata within an acceptable amount of time to ensure that the user experience is smooth and free of interruption.
- Useability and Space Requirements: The system shall have an emphasis on being a lightweight application that shall actively work to minimize the stress placed on the host operating system.
- Security Requirements: The system shall also encrypt user data to prevent any leak or any unauthorized access of user information.

Organizational Requirements:

The system shall properly function across different operating systems/devices. The system shall be operable in high stress situations including crowded networks and weak internet service. The development team shall use agile methodology to work efficiently ensuring that the system evolves to meet any changing needs/requirements from the users.

External Requirements:

The system shall comply with any regulations that are present in the educational institution or geographic location that the software is being used in. The system shall ensure that any student/teacher can interact with the system equally providing no advantages to any one group or student. The system shall comply with any law pertaining to data privacy, storage, or language of the system. The system shall maintain a log of user specific user interactions to ensure that any errors, task completion, grade changes are correct and authentic. Finally the system shall prevent any sharing of user data between multiple users or to outside unauthorized entities.

Project Scheduling

Start Date: 10/01/2024

End Date: 11/10/2024

Starting the project on the first of October gives our team approximately 6 weeks to plan and organize the development process, test each phase as it's developed (Prototyping Evolutionary Model), and execute product test trials. Our team decided that visualizing the development of our project in 3 broad phases would make it easier to schedule landmarks, while still allowing for simple/easy debugging. Our team is not scheduled to work on weekends, however we have reserved 3 weekend days towards the end of the timeline to give us extra time to compile and ensure our product is meeting our quality standards. Each member has agreed to (signed this document) work a minimum of 2 hours a week.

Cost, Effort, and Pricing Estimation:

Cost Model Employed: Function Point Analysis

Our team decided to use the Function Point cost modeling technique for our estimation report. Using this we were able to estimate the size and development costs for our project. FPA measures software size by quantifying its functionality from the user's perspective, providing a standardized approach to estimate development effort and cost. [1]

Identified Functional Components:

Student Use Cases:

1. View Personal Info
2. View Academic Standing
3. View Class Completion Progress
4. Contact Teacher
5. Complete Task/Assignment

Teacher Use Cases:

1. View Personal Info
2. View Class Analytics (Completion Progress per class)
3. View Class Analytics (Average GPA per class)
4. Contact Student
5. Create Task/Assignment
6. Edit Task/Assignment
7. pRemove Task/Assignment
8. Edit Specific Student's Metadata

Function Point Table:

Component Type	**Quantity**	**Complexity**	**Function Points per Component**	**Total Function Points**
Student Use Cases				
View Personal Info	1	Low	3	3
View Academic Standing	1	Average	4	4
View Class Completion	1	Average	4	4
Contact Teacher	1	Low	3	3
Complete Task/Assignment	1	High	6	6

Teacher Use Cases				
View Class Analytics (Progress)	1	Average	5	5
View Class Analytics (GPA)	1	Average	5	5
Contact Student	1	Low	3	3
Create Task/Assignment	1	High	6	6
Edit Task/Assignment	1	High	6	6
Remove Task/Assignment	1	Low	3	3
Edit Student Metadata	1	High	6	6
Total UFP				**54**

Adjusted Function Points: 54
We assumed a Value Adjustment Factor of 1.0 (average complexity), therefore the AFP remains the same

Effort Estimation:
Total Effort = 54 AFP x 5 hours (per Function Point) = **270 Hours**

Function	**Effort (Hours)**
Student Use Cases	
View Person Info	15
View Academic Standing	20
View Class Completion	20
Contact Teacher	15
Complete Task/Assignment	30
Teacher Use Cases	

View Class Analytics	25
Contact Student	15
Create Task/Assignment	30
Edit Task/Assignment	30
Remove Task/Assignment	15
Edit Student Metadata	30
Total Effort	**270 Hours**

Cost Estimation:

Personnel Cost

- The Average Hourly rate for Software Developers in the U.S. is around $41 [3]
- The Average Hourly rate for Software Developer I is $39 [5]
- Our team decided we would need experienced developers rather than the average developer therefore we used an assumed average Hourly Rate of $60 for Developers [4]
- Personnel Cost = 270 hours x $60 (Average Hourly Rate) = **$16,200**

Hardware Cost

- As the title of our project suggests, our product will use existing infrastructure to minimize hardware costs and allow for a smooth integration process.
- This cost will vary per institution therefore we took the higher average cost for new equipment (if needed) which was **$5,000**

Software Cost

- Our team will utilize open-source tools to try and minimize the software costs, however if licenses and certifications are needed this would add around $500 - $2000. Knowing this our team has decided to play it safe and assume the larger variation of **$2,000**

Training Cost

- Using popular industry reports we found that the average training cost for each employee had a range from $1000 - $1,500 [7]
- Our team was dedicated to preparing for the worst case scenario in every pricing estimation to avoid future surprises. Therefore our team has estimated a training cost of **$1,500**

Total Estimated Cost: $24,700

Component	**Cost (USD)**
Personnel	$16,200
Hardware	$5,000
Software	$2,000

Training	$1,500
Total Cost	**$24,700**

Product Pricing: $41,990

Based on the *Cost Estimation Report* our team has decided that our product pricing of $41,990 (following a 70% profitability margin) is fair for our development team, but at the same time keeps our product accessible to a larger majority of institutions. When deciding on a price for our project, our team had to weigh the average profitability margin for a SaaS product. According to our researched data, a good gross margin for a SaaS company is 75%+. [8] However since our project goal and motivation focus on providing students with a less stressful and more engaging way to learn, we decided a profit margin of 70% better aligns with our project scope, even if it is considered low.

Unit Test Plan

Verify the functionality of calculate_student_progress method:

This method should calculate and display a student's overall course completion progress using completed assignments and total assignments as inputted arguments. We will use the python library *unittest* to test this method. We will also code using the *Google Colab* environment, as it allows for sharing and editing across team members.

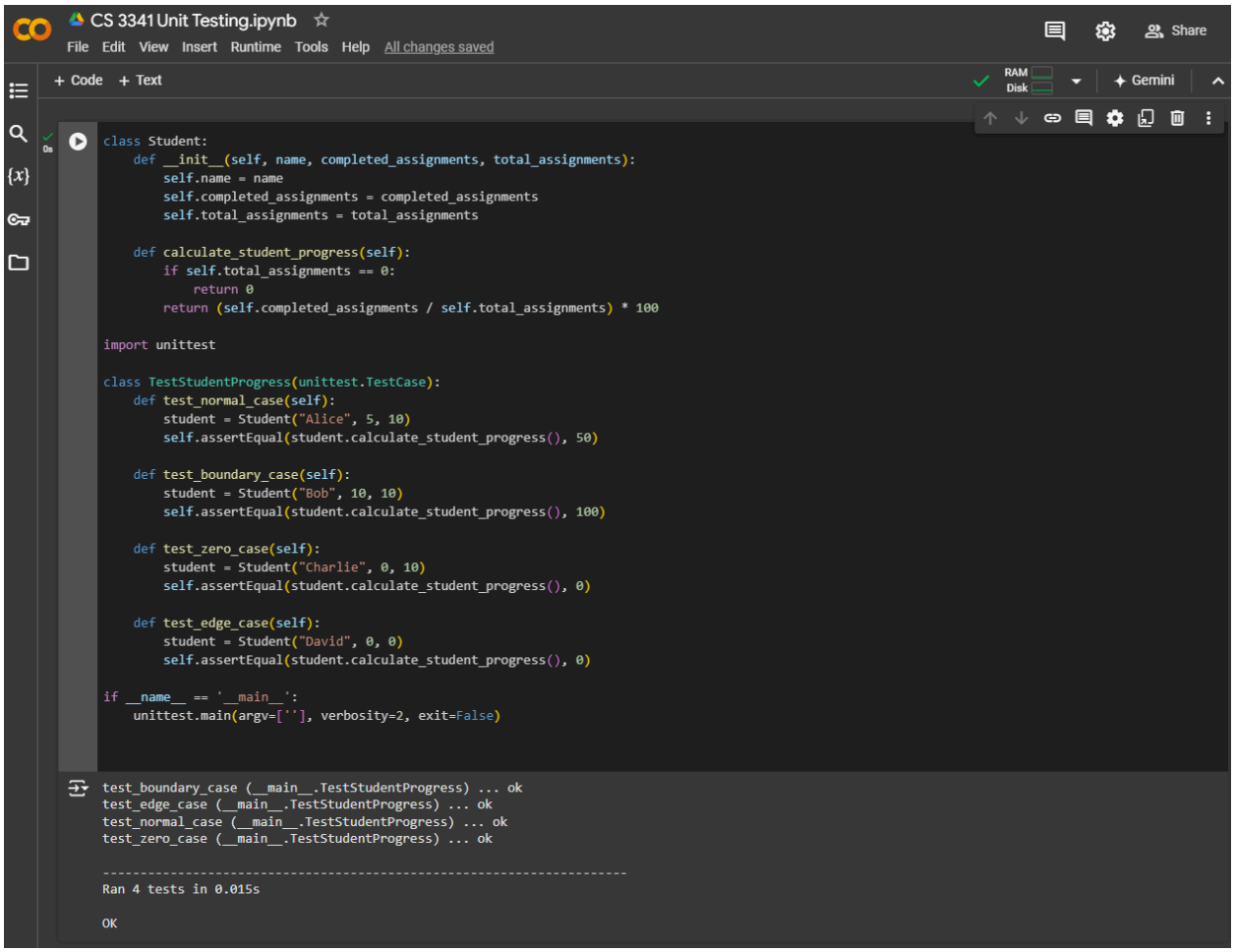
Test Cases:

1. Normal Case: The student completed some assignments
2. Boundary Case: The student completed all assignments
3. Zero Case: The student hasn't completed any assignments
4. Edge Case: There are no assignments assigned to the student (tests the possibility of division by 0)

Results:

1. Normal Case:
    a. *Input:* We created a scenario where 5 assignments were completed out of 10 total assignments
    b. Expected Output: 50%
    c. Result: ok (pass)
2. Boundary Case:
    a. *Input:* We created a scenario where 10 assignments were completed out of 10 total assignments
    b. Expected Output: 100%
    c. Result: ok (pass)
3. Zero Case:
    a. *Input:* We created a scenario where 0 assignments were completed out of 10 total assignments
    b. Expected Output: 0%
    c. Result: ok (pass)

4. Edge Case:
    a. *Input:* We created a scenario where 0 assignments were completed out of 0
    b. Expected Output: 0% (no errors as a result of division by 0)
    c. Result: ok (pass)



Products Similar to IVASI

Following our project goals and motivation, our team aims to provide students and institutions a way to gamify the world of academia. By doing so we hope to improve student satisfaction and engagement in class. We have found 3 products that already exists that are similar to our proposed project which are:

1. *ClassDojo*
    a. ClassDojo is a communication app for teachers, students, and families, focusing on building classroom culture through feedback and rewards. This app however focuses more on behavior tracking and communication than academic visualization. [9], [10], [11]
2. *Kahoot!*
    a. Kahoot! is a popular program that creates a competitive, game show-like environment for quizzes and other learning activities. While this program shares

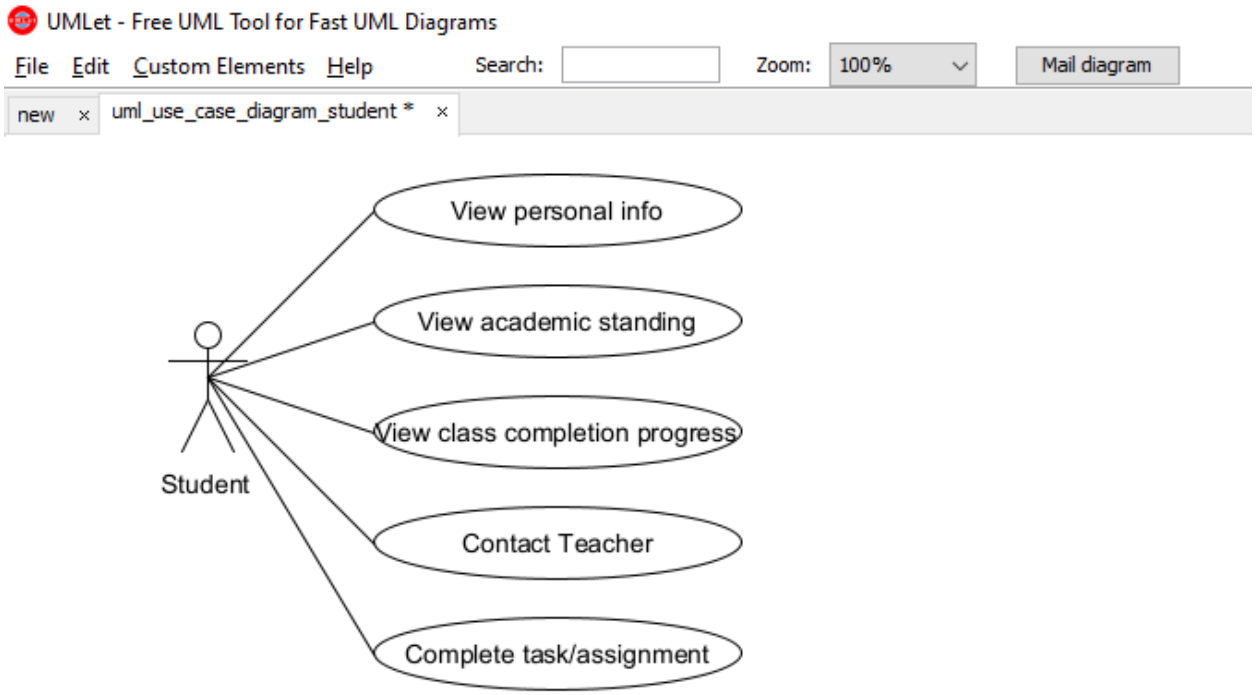
IVASI's goal of gamifying academic studies, it lacks the ability to track individual academic progress. [9], [10], [11]

3. *Quizlet*
    a. Quizlet is a platform that offers tools such as flashcards and games to aid students in their studies. While this program is a great way for students to get engaged it lacks the capability to track overall academic progress and provide individual specific feedback. [9], [10], [11]

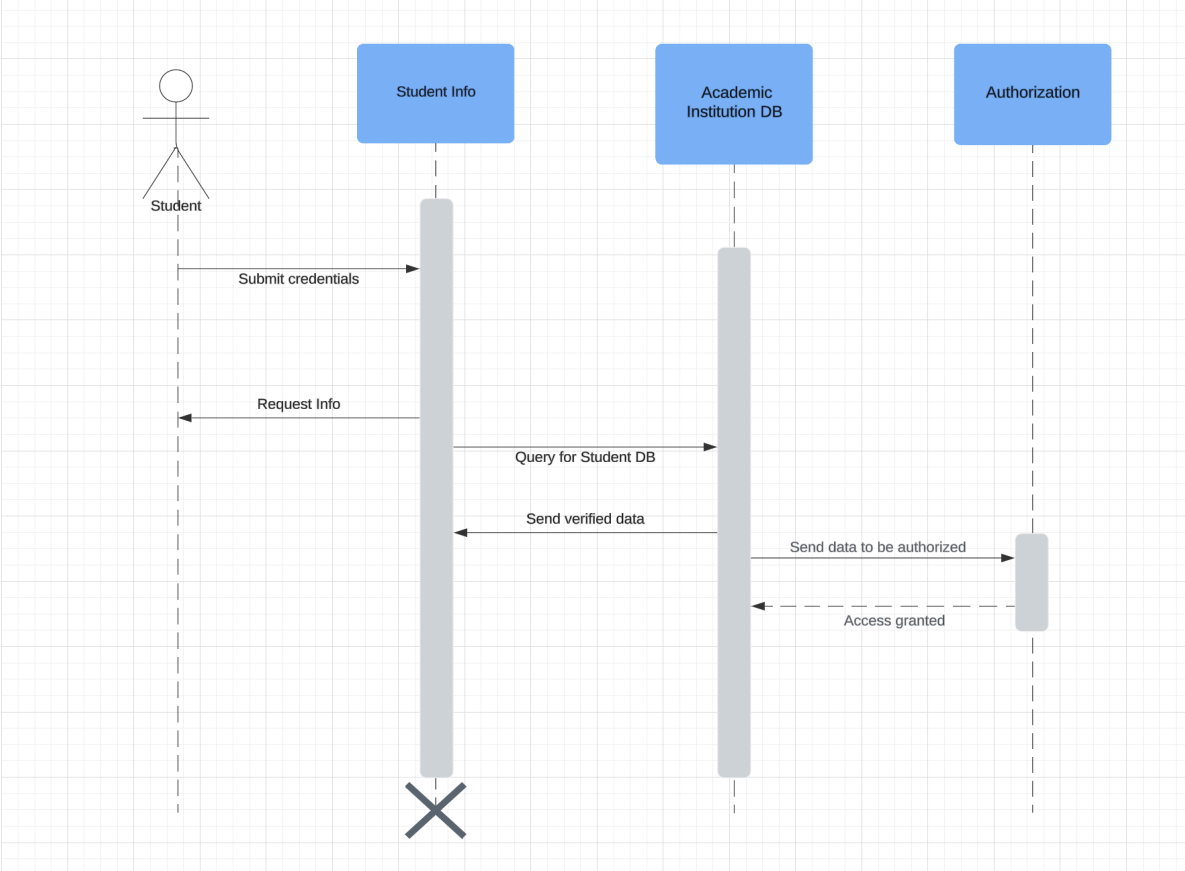
Project Summary:

Throughout this project's development process we were able to stick close to our original project scope, *Project Goal*, *Project Motivation*, and other outlined requirements. We believe this is a positive testament to our group's ability to effectively communicate which allowed us to adapt to and overcome any unforeseen complications. We did have to make a few changes to the following diagrams: *Sequence Diagram (View Personal Info)*, *Class Diagram*, and *Architectural Design (Repository Architecture Pattern)*. We made these changes because we were made aware of a few mistakes in our documentation format, however we were able to fix these mistakes to more accurately and correctly convey our project scope. We also had to make a few changes to our *Functional Requirements* and *Non-Functional Requirements*. There were instances where we were not specific enough in our documentation which could have led to potential legal problems in the future. Therefore we made sure that our documentation specifically and explicitly conveyed our desired project scope and goals. In conclusion, we are pleased to report that our team was able to effectively create a project design plan, execute the design plan, and fix any mistakes that came up along the way and end up with a stable presentable product.

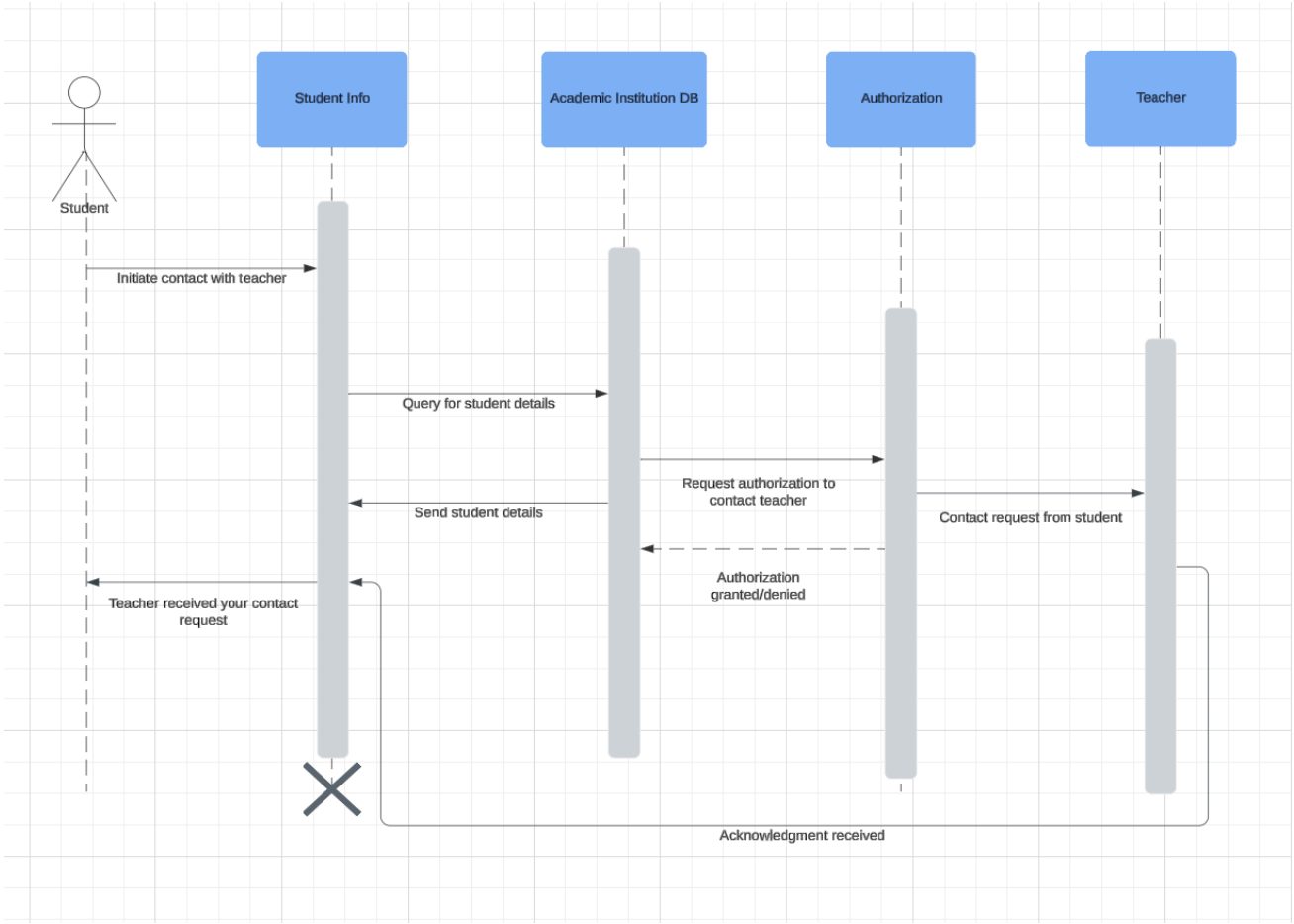
Use Case Diagram


UMLet - Free UML Tool for Fast UML Diagrams
File Edit Custom Elements Help
Search:
Zoom:
100%
Mail diagram
new
uml_use_case_diagram_student *
View personal info
View academic standing
View class completion progress
Contact Teacher
Complete task/assignment
Student

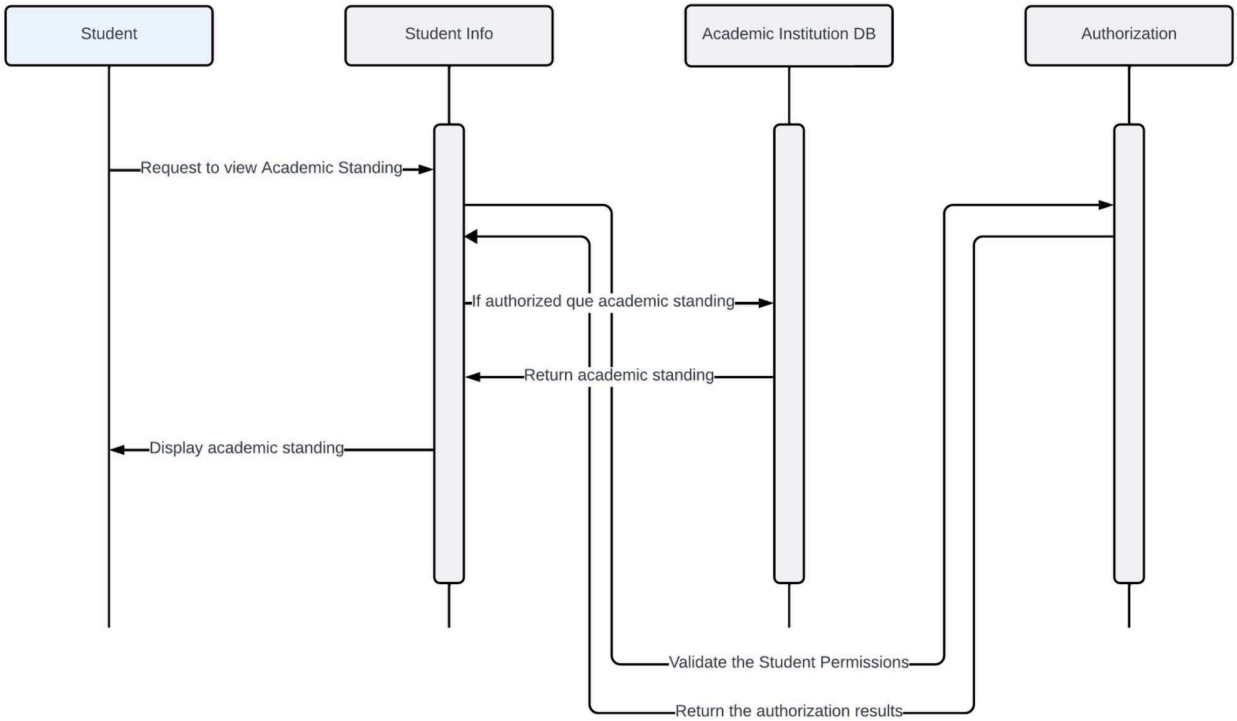
Sequence Diagram (View Personal Info)



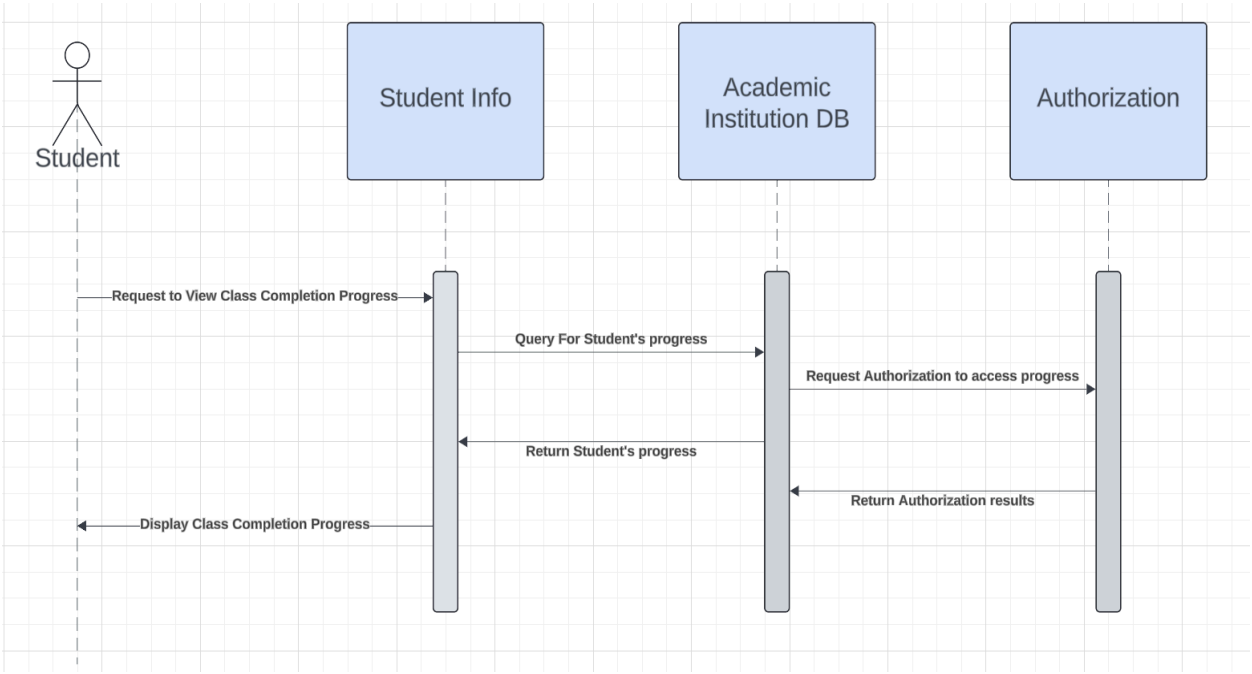
Sequence Diagram (Contact Teacher)



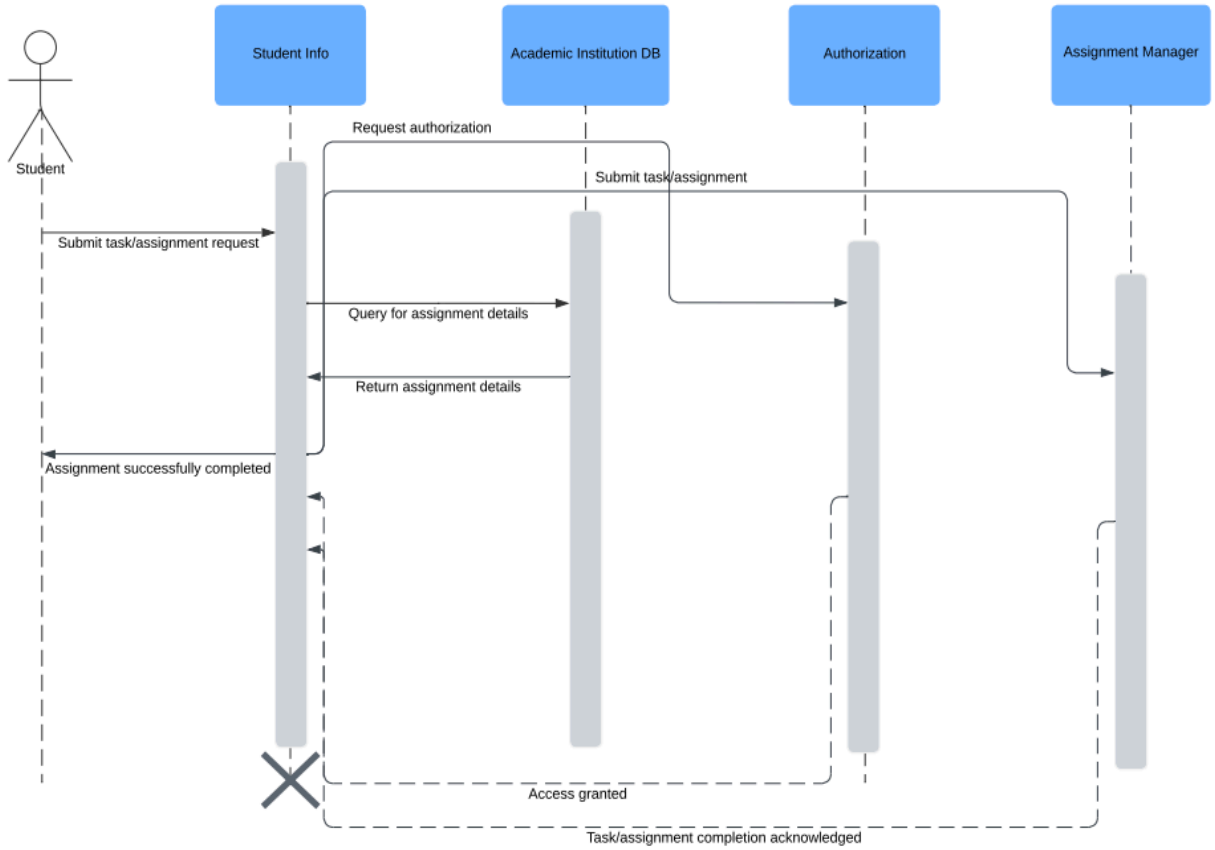
Sequence Diagram (View Academic Standing)



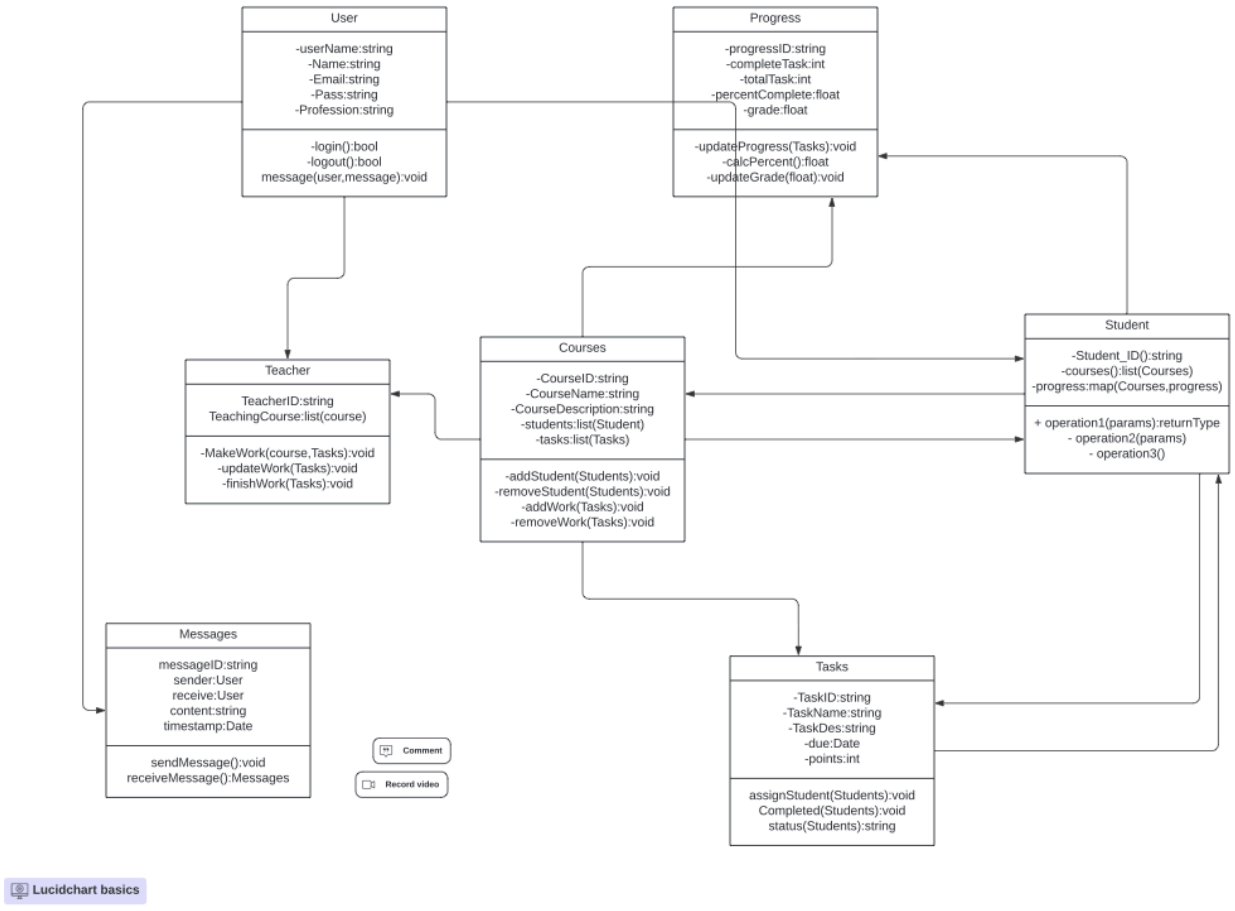
Sequence Diagram (View Class Completion Progress)



Sequence Diagram (Complete Task/Assignment)

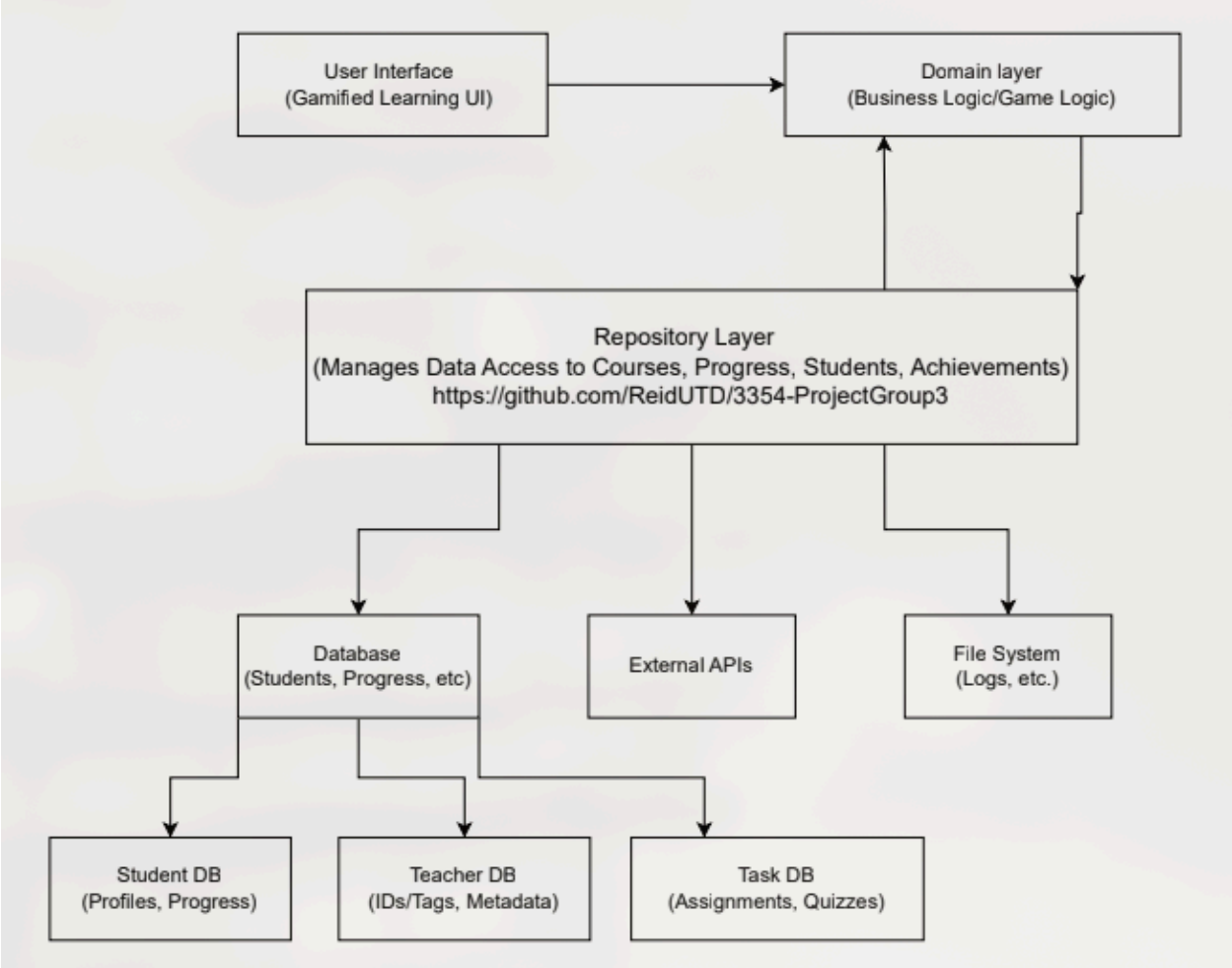


Class Diagram



Lucidchart basics

Architectural Design (Repository Architecture Pattern)



References

1. Fingent, "Function Point Analysis – Introduction and Fundamentals," [Online]. Available: https://www.fingent.com/blog/function-point-analysis-introduction-and-fundamentals/. [Accessed: 04-Nov-2024].
   This source provides an overview of Function Point Analysis (FPA), explaining its purpose and methodology in estimating software size and development effort.
2. GeeksforGeeks, "Functional Point (FP) Analysis – Software Engineering," [Online]. Available: https://www.geeksforgeeks.org/software-engineering-functional-point-fp-analysis/. [Accessed: 04-Nov-2024].
   This article offers a detailed explanation of Function Point Analysis, including its calculation methods and applications in software engineering, which aids in understanding the estimation process.
3. Payscale, "Hourly Rate for Industry: Software Development," [Online]. Available: https://www.payscale.com/research/US/Industry%3DSoftware_Development/Hourly_Rate. [Accessed: 04-Nov-2024].
   This source offers data on average hourly rates for software developers in the United States, which is essential for calculating personnel costs in the estimation.
4. CloudDevs, "Hourly Earnings Unpacked: The Ultimate Software Developers Rate Guide," [Online]. Available: https://clouddevs.com/software/hourly-rates/. [Accessed: 04-Nov-2024].
   This article provides insights into the average hourly wages of software developers in the USA, highlighting variations based on experience levels, which aids in refining personnel cost estimates.
5. Salary.com, "Hourly Wage for Software Developer I," [Online]. Available: https://www.salary.com/research/salary/alternate/software-developer-i-hourly-wages. [Accessed: 04-Nov-2024].
   This source details the average hourly wages for entry-level software developers, offering a range that helps in determining appropriate compensation rates for personnel.
6. U.S. Bureau of Labor Statistics, "Software Developers," [Online]. Available: https://www.bls.gov/oes/current/oes151252.htm. [Accessed: 04-Nov-2024].
   This official government source provides comprehensive data on employment and wage statistics for software developers across various industries and regions in the United States, serving as a reliable benchmark for salary estimations.
7. *EZO.io, "4 Successful Strategies for Managing the Cost of Software Licenses," [Online]. Available: https://ezo.io/assetsonar/blog/cost-of-software-licenses/. [Accessed: 04-Nov-2024].*
8. chargebee.com, “What is SaaS Gross Margin?” [Online]. Available: https://www.chargebee.com/resources/glossaries/saas-gross-margin/. [Accessed: 04-Nov-2024
9. EdTech Empire, "Top 10 Gamification Apps for Education," [Online]. Available: https://edtechempire.com/top-10-gamification-apps-for-education/. [Accessed: 04-Nov-2024].

10. ClassPoint, "Top 7 Gamification Tools for Teachers to Boost Classroom Engagement," [Online]. Available: https://www.classpoint.io/blog/top-7-gamification-tools-for-teachers-to-boost-classroom-engagement. [Accessed: 04-Nov-2024].
11. Jotform, "Top 10 Gamification Apps for Education," [Online]. Available: https://www.jotform.com/blog/gamification-apps-for-education/. [Accessed: 04-Nov-2024].