# Interactive and Visual Academic Standing Integration:

## Deliverable 2

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

# Project Goal

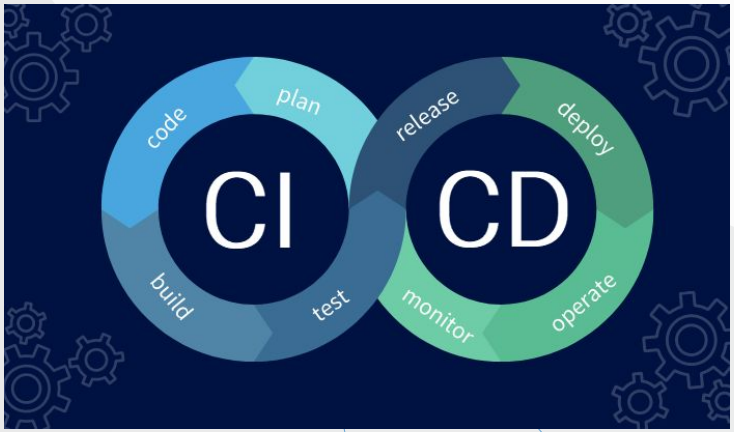
- Implementing the engineering design process to create a program that can be seamlessly integrated into a classroom learning environment
  - Convert a course's assignments, quizzes, and other related tasks into a video game-like format.
  - The aim is to design a program that would help students stay engaged with the content as well as visualize their academic progress/standing in the class.

# 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.
    - 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.

# Software Process Model

- Our group has chosen to implement the Prototyping Evolutionary Model.
  - 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.
- 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.



# Functional Requirements

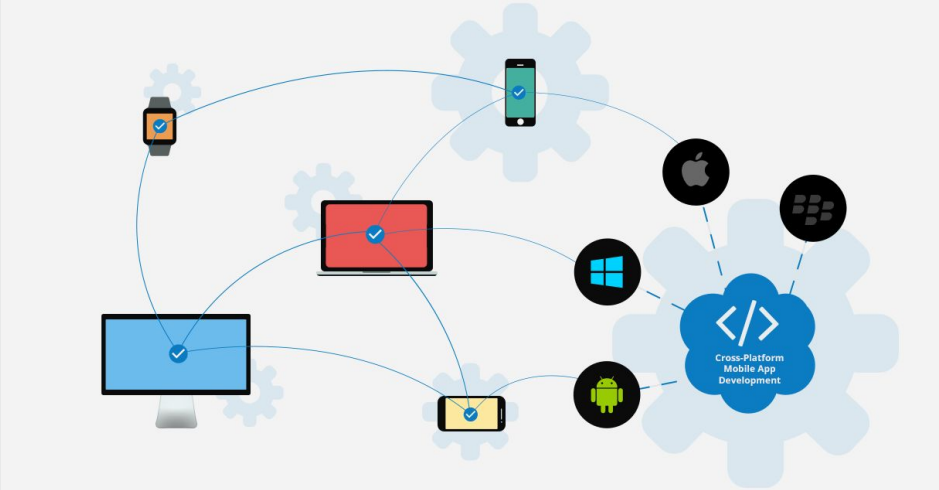
- A user shall be able to query any/all parts of their profile whenever necessary.
- The system shall generate a user's (student's) metadata for the user to view using graphs, progress bars, large buttons, and minimal text.
- Each teacher/professor using the system shall be uniquely identified by a particular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

# Non-Functional Requirements

- Product Requirements
  - Dependability Requirements: The system shall be available to any educational organization and fully functional during workday hours, which means it 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 time to ensure the user experience is smooth and interruption-free.
- Usability and Space Requirements: The system shall emphasize being a lightweight application that actively works to minimize the stress on the host operating system.
- Security Requirements: The system shall also encrypt user data to prevent leaks or unauthorized access to user information.

# Non-Functional Requirements

- 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 the users' changing needs/requirements.



# Non-Functional Requirements

- External Requirements
  - The system shall comply with any regulations present in the educational institution or geographic location in which the software is being used. 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 about data privacy, storage, or language. The system shall maintain a log of user-specific user interactions to ensure that any errors, task completion, and grade changes are correct and authentic. Finally, the system shall prevent any sharing of user data between multiple users or outside unauthorized entities.

# Project Scheduling

Project Timeline

- Start Date: 10/01/2024
- End Date: 11/10/2024
- Project Duration: ~6 weeks for full project cycle, including planning, development, testing, and product trials.
- Development Model: Prototyping Evolutionary Model
  - Testing each phase as it's developed to refine and improve iteratively.
- Project Phases: Visualized in 3 broad phases to:
  - Schedule key milestones effectively.
  - Allow for easier debugging and adjustments.
- Work Schedule:
  - Team members are not scheduled to work on weekends.
  - 3 weekend days reserved near the end of the timeline for final compilation and quality assurance.
- Time Commitment:
  - Each team member has signed to commit a minimum of 2 hours per week.

# Cost, Effort, and Pricing Estimation

Cost Model Employed: Function Point Analysis (FPA)

- Chosen Technique: Function Point Analysis (FPA) for cost estimation.
- Purpose:
  - To estimate the size and development costs of the project.
  - To provide a standardized approach for assessing development effort and cost.
- How It Works:
  - FPA measures software size based on its functionality from the user's perspective.
  - Quantifies functionality to create an objective, user-centered estimate.

https://www.rawpixel.com/search/public%20domain%20money?page=1&path=_topics&sort=curated

# Cost, Effort, and Pricing Estimation

*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

https://www.freepik.com/free-photos-vectors/school-students

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. Remove Task/Assignment
8. Edit Specific Student's Metadata

https://www.stockvault.net/photo/254580/zlib/trees

# Cost, Effort, and Pricing Estimation

*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

# Cost, Effort, and Pricing Estimation

Total Estimated Cost: $24,700

| Component | Cost (USD) |
|---|---|
| Personnel | $16,200 |
| Hardware | $5,000 |
| Software | $2,000 |
| Training | $1,500 |
| **Total Cost** | **$24,700** |

# Cost, Effort, and Pricing Estimation

## 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. Still, at the same time, it keeps our product accessible to a more significant 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, an excellent 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 that a profit margin of 70% better aligns with our project scope, even if it is considered low.


OpenAI. (2024). *A professional scene depicting a team meeting about product pricing for an educational software project* [AI-generated image]. DALL-E. Available upon request in ChatGPT.

# Unit Test Plan

We will use the python library *unittest* to test this method. We will also code using the *Google Collab* environment, as it allows for sharing and editing across team members.

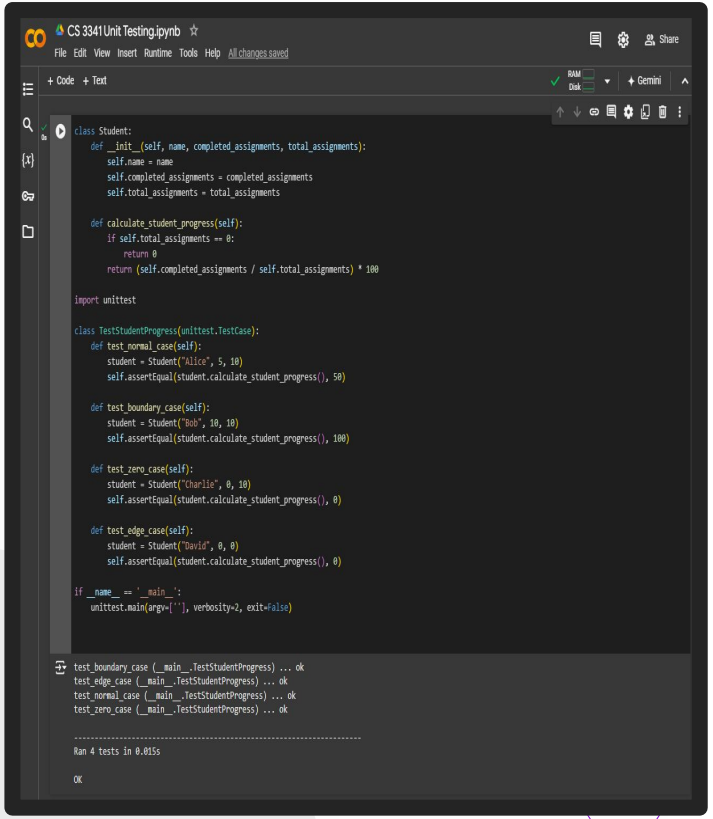
*Test Cases:*

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)

# Unit Test Plan

*Test Cases and 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)



# Similar Products

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]
4. *Edmodo*
    a. Edmodo is a more comprehensive learning management system (LMS) that incorporates assignment tracking, grading, and messaging between teachers and students. Although not heavily gamified, it supports classroom engagement through interaction and feedback. [9], [10], [11]

# Project Summary

**Adherence to Project Scope**: Our team maintained alignment with the original project scope, goals, and outlined requirements, ensuring we stayed true to the project's intended direction.
**Effective Team Communication**: Strong communication within the team allowed us to quickly adapt and respond to unforeseen challenges, leading to a smoother development process.
**Diagram Adjustments**:

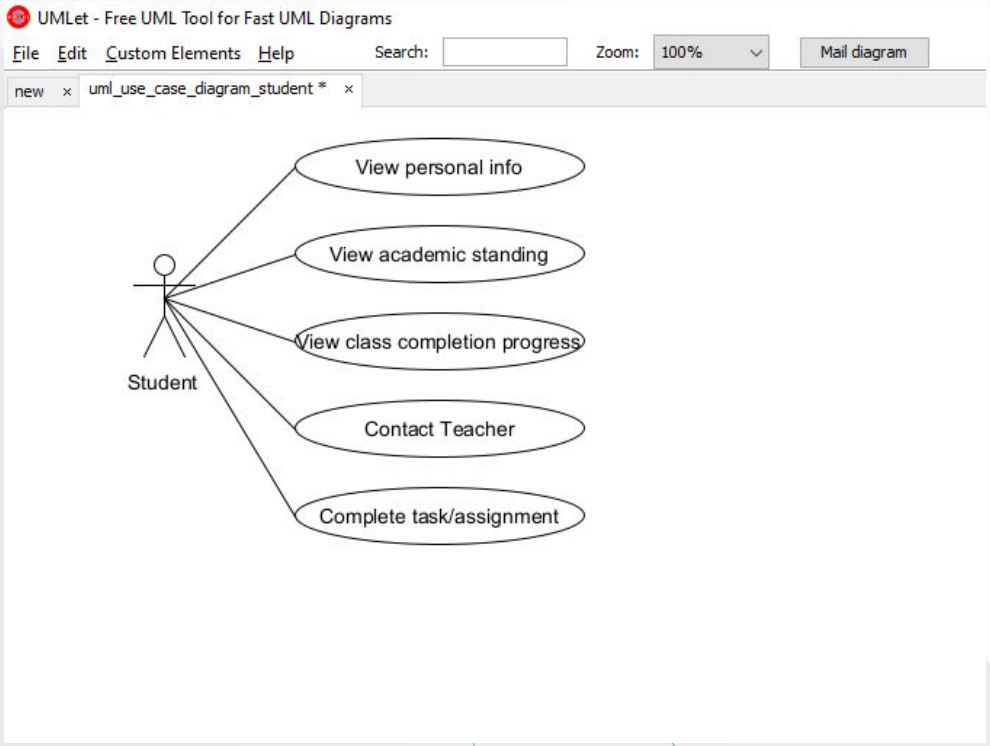
- **Sequence Diagram (View Personal Info)**: Modifications were made to improve clarity and correct errors in documentation.
- **Class Diagram**: Revised to reflect a more accurate representation of relationships and interactions within the system.
- **Architectural Design (Repository Architecture Pattern)**: Adjusted to enhance system design, format flexibility, cross platform support and align with best practices.

**Requirement Revisions**:

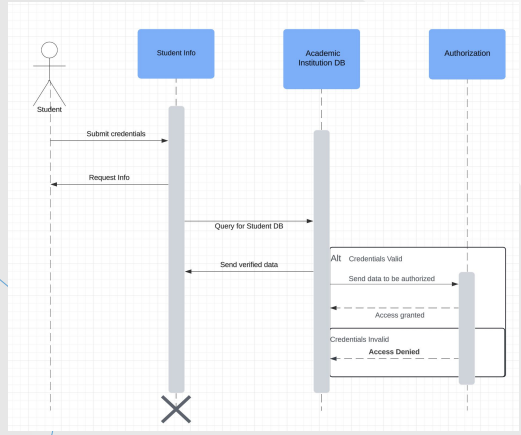
- **Functional and Non-Functional Requirements**: Documentation was refined to enhance specificity, addressing potential ambiguities that could lead to legal or operational issues. This ensured a clear and accurate conveyance of our project's scope and objectives.

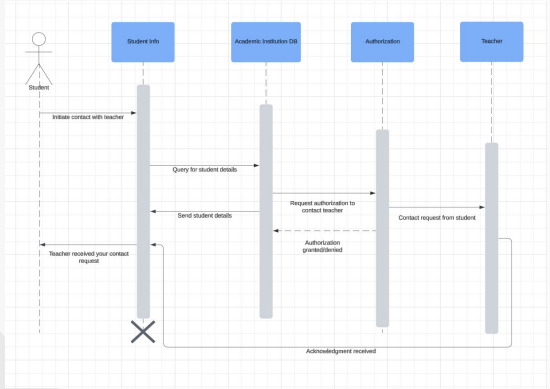
**Outcome**: We successfully designed, executed, and iterated on the project plan, effectively addressing any issues along the way, resulting in a stable, presentable product that meets the project's goals.

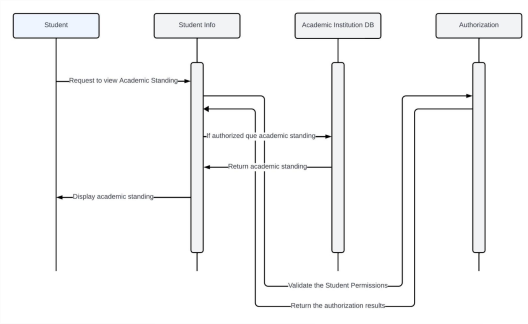
# Use Case Diagram

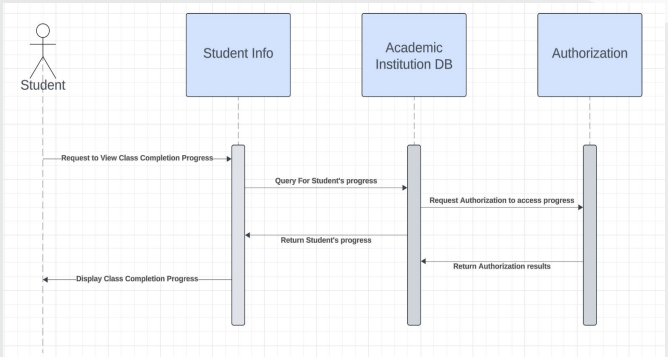


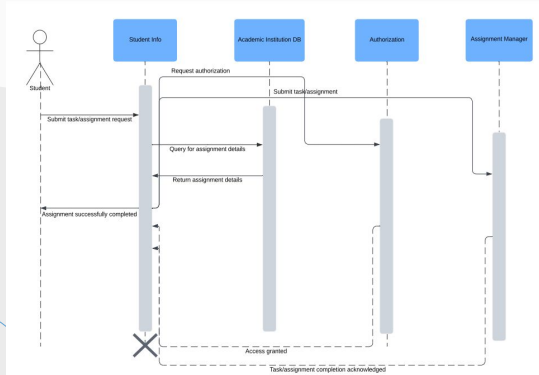
# Sequence Diagrams



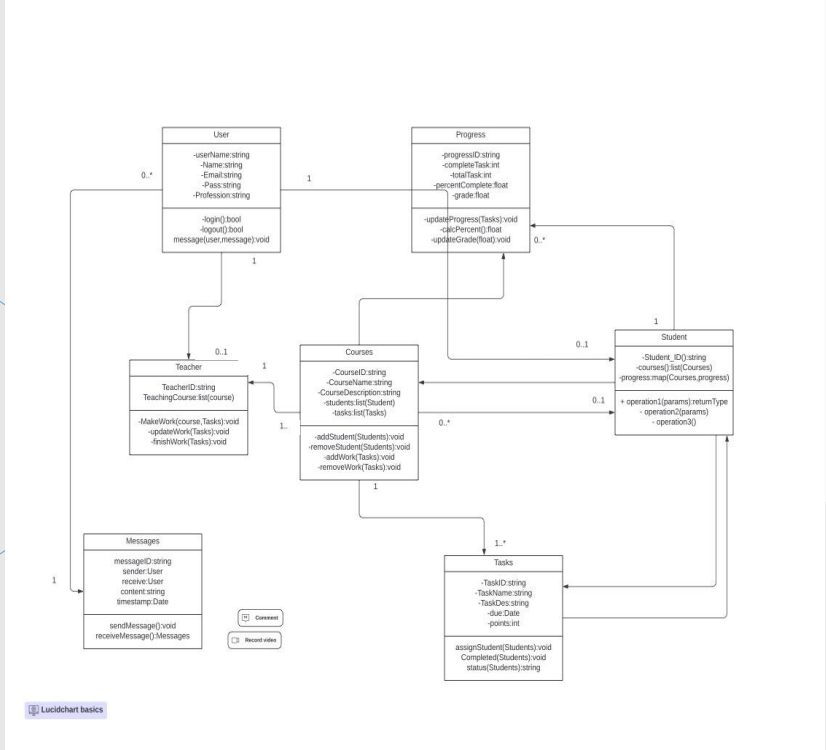



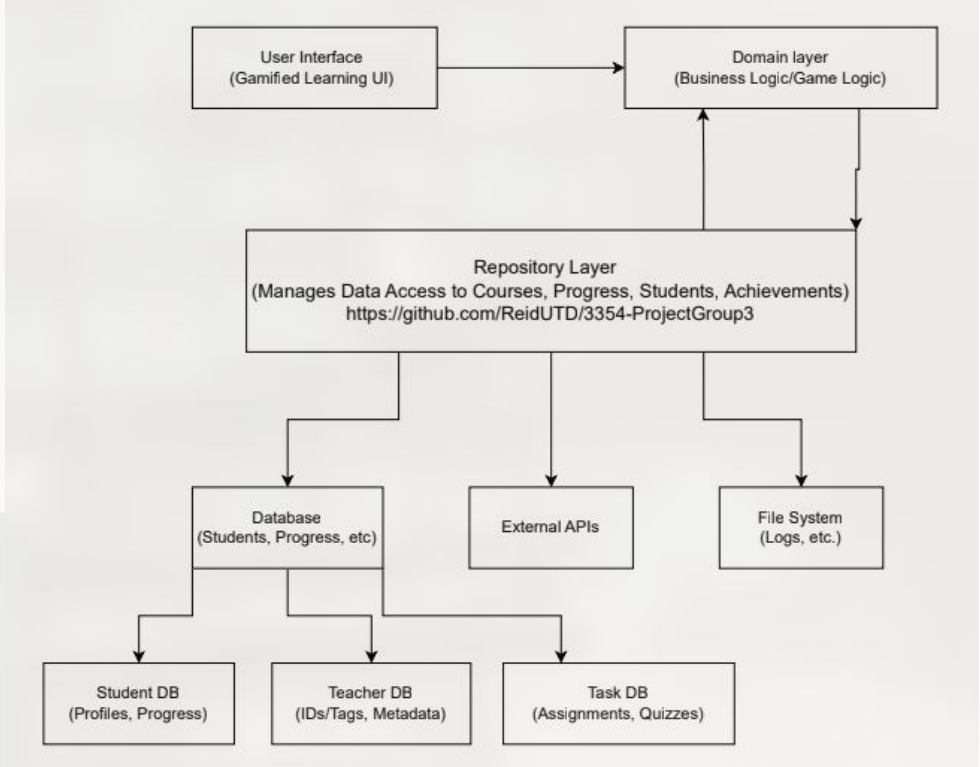



# Class and Architectural Diagram



# 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].
12. H. Asal, “What is Cross-Platform Software? - Hakan Asal,” Medium, Jul. 17, 2019. Accessed: Nov. 10, 2024. [Online]. Available: https://medium.com/@hakanasal51/what-is-cross-platform-software-38ee57b7304a
13. G. Contributor, “5 Benefits of Implementing a CI/CD Pipeline — Ranorex Blog,” Ranorex. Accessed: Nov. 10, 2024. [Online]. Available: https://www.ranorex.com/blog/5-benefits-ci-cd-pipeline/