Changes Impact Report

ECE 651 Team #6

Architecture

The architecture of this project has been changed from an independent app to a client/server architecture for the main part with two more apps for course management and user management respectively.

- By splitting the functionality into multiple apps, the project becomes more modular and easier to maintain. Each app can focus on a specific set of features, leading to better code organization and easier debugging.

- The client/server architecture can improve performance by offloading processing tasks to the server, reducing the workload on client devices.

- Data is not specific to a machine any more, client application run on different machines share the data from server side.

Data Storage - RDBMS

The method of data storage has been changed from storing data in an encrypted file to using a remotely connected Relational Database Management System (RDBMS).

- RDBMS systems can easily integrate with other systems and applications, making it easier to share data and collaborate across different parts of an organization or ecosystem.
- RDBMS systems provide built-in mechanisms for managing concurrent access to data and enforcing access control policies. This ensures that multiple users can safely access and manipulate data without risking data corruption or unauthorized access.
- RDBMS systems typically offer robust backup and recovery mechanisms, allowing for the creation of regular backups and the restoration of data in the event of data loss or corruption.
- RDBMS systems ensure data integrity and consistency by enforcing ACID (Atomicity, Consistency, Isolation, Durability) properties. This means that transactions are reliably processed, ensuring that data remains accurate and reliable.
- RDBMS systems are designed to handle large volumes of data and can scale vertically or horizontally as needed. This allows for the storage and management of increasing amounts of data over time.

Data Structure

User/Account

Previously, User was an interface for any person(faculty or student) in this system, while account was a interface for anyone who could log in to this system. That is, every account belonged to a single person, but not every person had an account. After discuss with the user, the final decision is every single person has their own account, so these two interface are combined to be the updated “user” interface.

- Combining the "User" and "Account" interfaces into a single, updated "User" interface simplifies the system architecture and reduces complexity. This consolidation streamlines the user experience by providing a consistent interface for all individuals interacting with the system.

- With every person having their own account, it becomes easier to track and attribute actions within the system to specific individuals. This enhances accountability and auditability, which are crucial for regulatory compliance and troubleshooting purposes.

Course/Section/Enrollment

Previously, there was no section concept in this system. Besides, the list of all student identifiers of students who enrolled in the course was stored in the course. Current section concept is similar to course in the previous version, but a course can have multiple sections. For the updated version, with the Database utilized, whether a student has enrolled in a section can be queried from enrollment table in database instead of stored in the section or course one.

- The introduction of the section concept enhances the granularity and flexibility of the system. Sections allow for better organization and management of courses, enabling administrators and instructors to tailor the learning experience to specific groups of students or cohorts.

- Storing the enrollment data in a separate table in the database represents a normalization of the data structure. This improves data integrity, reduces redundancy, and enhances the efficiency of data management and querying processes.

- Separating enrollment information into its own table in the database simplifies data maintenance tasks. Updates to student enrollments, such as adding or removing students from a section, can be performed more efficiently and with less impact on other parts of the system.