ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

Project Proposal Report

SWE 4304: Software Project Lab I

Project Title

IUT CafeCache

Team Members

Mahmudul Hossain Samir (230042101)
Tamjidur Rahman (230042109)
S.M. Raiyan Al Arafat (230042134)

Supervisor

Sabbir Ahmed
Assistant Professor

Table of Contents

1. Project Overview

The IUT CafeCache is a Cafe Management System which is mainly designed as a learning exercise to teach our team effective group collaboration and the techniques required to structure and manage a large software project within specific technical constraints.

Technical Constraints and Architecture:

The system maintains a strict set of technical constraints:

- **Programming Language:** The entire system, the IUT CafeCache CORE (Backend layer), is implemented exclusively in C++.
- **Interface:** The application is strictly terminal-based (CLI). All user interactions are handled via text input and output in the console.
- **Library Usage:** The limit to library functions is set to 5. Thus, it emphasises the development of foundational programming skills and manual implementation of file handling and basic UI logic.
- **Data Persistence:** The system utilises a File-Based Database approach, with all persistent data (Users, Menu, Tokens, Logs) stored in local files.

Functional Scope:

The system is designed around two primary user roles:

- **Admin (Provost Office/Staff):** Focuses on operational management, including Menu Management and Financial Oversight (Sales Reports, Recharge Approval).
- **Normal User (Student/Teacher):** Focuses on transactional activities, including viewing the menu, utilising the internal E-wallet, and generating E-tokens for purchases.

This project serves as a practical demonstration of applying theoretical Software Engineering principles by creating a functional and real world project which we can use in our everyday life.

2. The Motivation Behind the Project

The IUT CafeCache project is driven by two main motivations. One is addressing operational inefficiencies, and the other is providing our team with hands-on experience in a large project.

User Centric and Operational Motivation

The primary motivation is to improve the existing cafe system by solving issues.

- **Addressing Inefficiency:** The current model suffers from long queues and cash handling issues, leading to frustration and delays.
- **Digitising Workflow:** CafeCache introduces a digital workflow to replace manual processes like the use of physical tokens.
- **Enhancing Management Insight:** The existing system does not have real time insights which is necessary for effective decision making later. This system implements a Reporting & Analytics module within the IUT CafeCache CORE, which provides Admin with data on sales and the menu.
- **Streamlining Transactions:** Wallet & Transaction Handling system enables the generation of E-tokens, which helps the system to be faster, more secure.

Developer and Learning Motivation

The secondary motivation focuses on the educational and professional development goals of our development team.

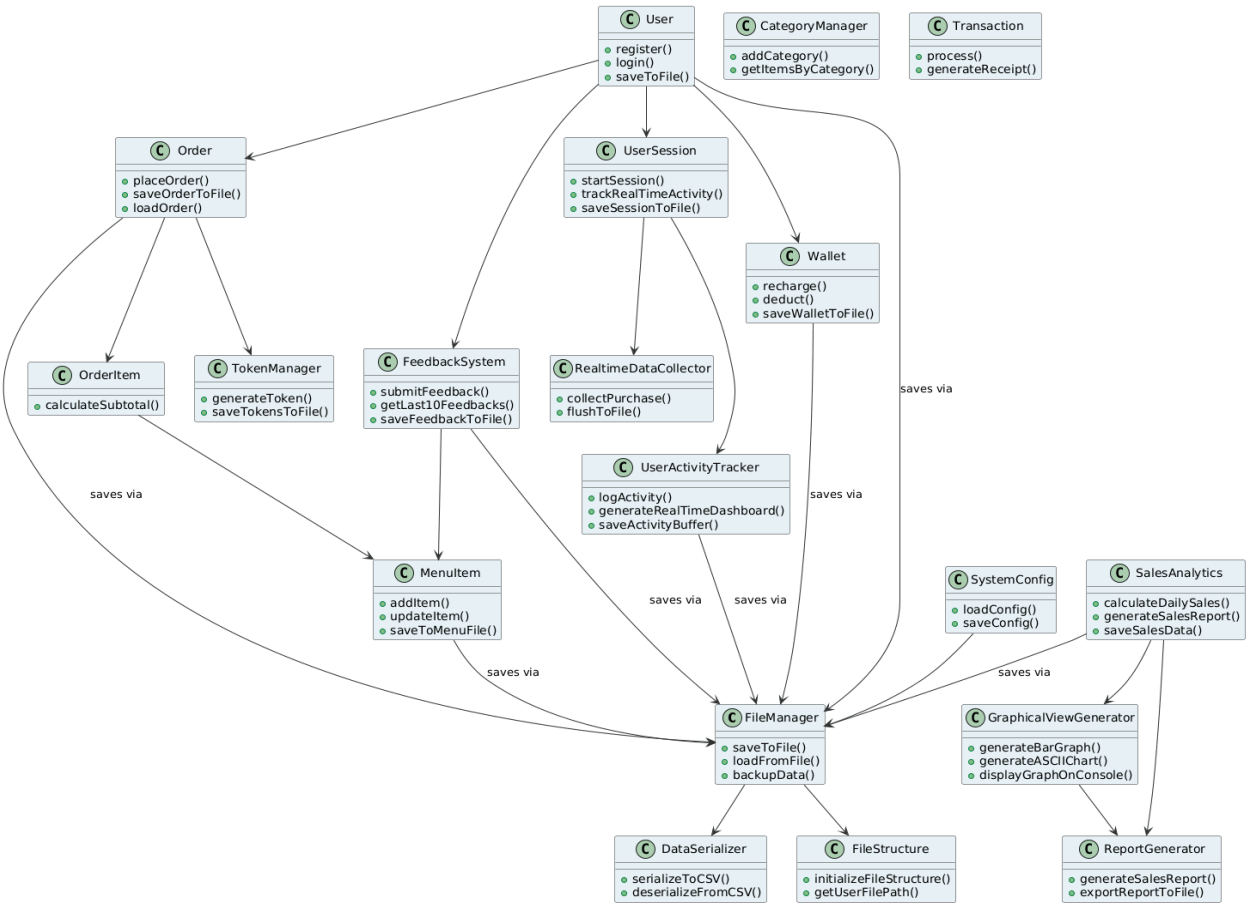
- **Hands on Experience:** Building this system from scratch provides the team with hands-on experience in understanding and executing a complete backend workflow.
- **Core Development Focus:** The project necessitates implementing complex features like Data Processing, Validation & Authorization, and File Storage in C++, ensuring a deep understanding of core programming and data management principles.
- **Collaborative Practice:** This project serves as a practical exercise in managing a large software project within a group. It requires version control (Git and GitHub) to ensure project cohesion and code quality.

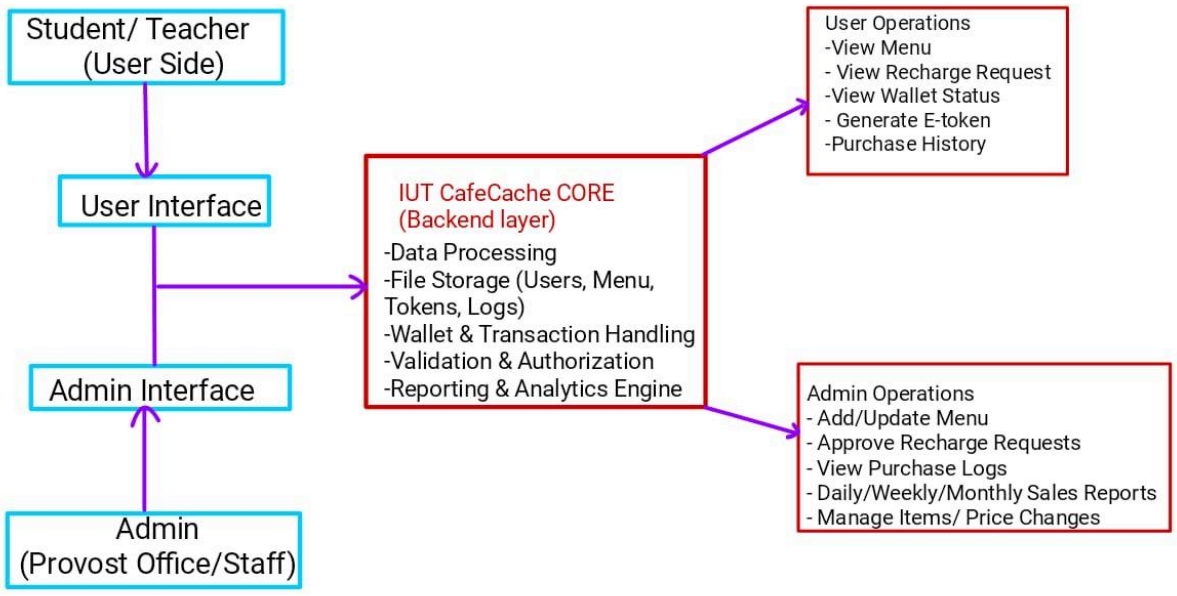
3. UML Class Diagram & the System Workflow

The UML class diagram represents a high-level structural design of the IUT CaféCache system. It focuses on core backend components such as user management, order processing, wallet handling, token generation, and sales analytics.

The system follows a modular design approach where each class has a clearly defined responsibility.
File persistence is centralized through the FileManager and DataSerializer classes to ensure consistency and maintainability.

This diagram is intentionally kept minimal at the proposal stage to emphasize system architecture rather than implementation details. This class diagram only outlines the workflow. It does not represent the whole part of the project.
The design is flexible and will be gradually enriched during development with additional attributes, validations, and refinements based on evolving requirements.





4. Key Features

The system's features are categorised based on the user type they serve:

A. User Operations (Student/Teacher)

These features are designed to manage ordering, transactions, and account information for the Normal User.

- **View Menu:** Allows users to browse all available items and prices.
- **View Wallet Status:** Displays the user's current wallet balance.
- **View Recharge Request:** Shows the status of the submitted E-wallet recharge requests.
- **Order food:** Creates a unique electronic token for ordering items.
- **Purchase History:** Provides a detailed log of all previous transactions and orders.

B. Admin Operations (Provost Office/Staff)

These features provide the tools necessary for operational and financial management:

- **Add/Update Menu:** Enables adding new items or modifying existing menu details and prices.
- **Approve Recharge Requests:** Allows administrators to process and approve/reject user wallet recharge requests.

- **View Purchase Logs:** Provides access to the summary record of all transactions.
- **Daily/Weekly/Monthly Sales Reports:** Generates data reports for analysing sales, revenue, and trends by using graphs.
- **Manage Items/Price Changes:** Allows administrators to manage item inventory and update prices.

5. Tools and Technology

The development of the IUT CafeCache system utilises a set of tools. These tools assist our team to be well equipped to build the proposed software.

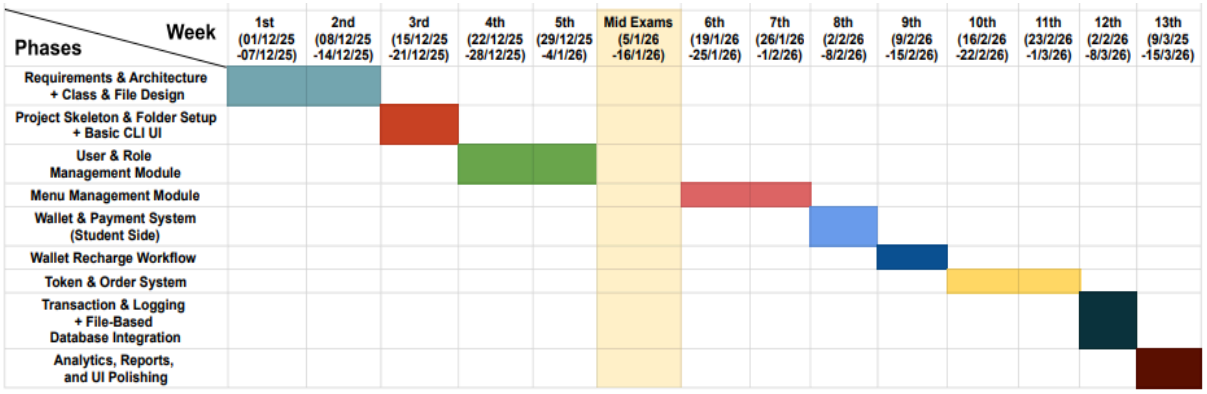
- **Programming Language (C++)**: Chosen for its performance in handling complex data processing and fast execution time.
- **Development Environment (VS Code)**: This provides a versatile environment for coding, debugging, and efficient management of the C++ project files.
- **Version Control (Git)**: Used for local version control, allowing the team to track changes and manage the history of the codebase.
- **Version Control Hosting (GitHub)**: Serves as the remote repository for collaboration, code backup, and managing the project's evolution across the team.
- **Testing (Google Test Framework)**: Employed to implement unit and integration tests, ensuring the correctness of core functions within the C++ backend.

  ****After discussion, we have decided not to use Google Test Framework as it can be considered redundant.*

- **Documentation (Google Docs)**: It enables collaborative report writing, project documentation, and maintaining records of requirements and architectural decisions.

6. Proposed Timeline

The project timeline is organized over 13 weeks and is structured to address dependencies, starting with architectural foundation and concluding with final analytics integration and UI polishing.



Phase	Duration	Activities and Deliverables
Requirements & Architecture	1st and 2nd week	Define requirements, establish system architecture, and finalize Class & File Design.
Project Skeleton & Setup	3rd week	Set up the initial Project Skeleton, Folder Structure, and implement the Basic CLI UI.
User & Role Management	4th and 5th week	Develop the module for managing different users (Student/Teacher and Admin).
Menu Management Module	6th and 7th week	Implement the Admin functionality for Add/Update Menu and Manage Items/Price Changes.

Wallet & Payment System (Student)	8th week	Develop the core Wallet & Payment System functionality on the User Side, enabling features like View Wallet Status.
Wallet Recharge Workflow	9th week	Implement the full Recharge Workflow, including the User's View Recharge Request and the Admin's Approve Recharge Requests.
Token & Order System	10th and 11th week	Implement the core ordering mechanism, including the Generate E-token feature.
Transaction & Logging	12th week	Develop robust Transaction Logging and integrate the File-Based Database. This underpins Purchase History and View Purchase Logs.
Analytics, Reports, & UI Polishing	13th week	Finalize the Reporting & Analytics Engine to generate Daily/Weekly/Monthly Sales Reports and polish the overall user interfaces.

7. Action Taken Based on Feedback & Plans for Changes

Feedback review led to two actions: adjusting the project delivery plan and enhancing system functionality.

Presentation Time Management: Our team is now focused on timing and practice schedule to ensure the next presentation is completed within the given time.

Implementation of a 'Feedback' Feature: A dedicated Feedback feature has been suggested to be integrated into the User Interface. Which allows Normal Users to submit comments and suggestions directly to the system. But, due to limited time and complexity we have decided to keep it optional for now. This is due to the fact that other implementations may take much time to solve.
If it is possible to finish the simple phase of the application ahead of scheduled delivery; including the ‘Feedback’ feature we plan to add ‘Refund System’, and ‘Popular Item’ features too.