Group Assignment

Project Proposal

Group 4

Course: BSc (Hons) in Contemporary Software Development

Module: SWDE_IT803 - LY_ICSWD_B: Software Development (2025/26)

Lecturer: Lusungu Mwasinga

Submission Date: 18.10.2025

Table of Contents

1. Team Information

Team Name – **The Fantastic (Group 4): First Steps**

Student Id	First Name	Last Name	Participation
L00196579	Chris	Byrne	100%
L00196830	Francesco	Sciabbarrasi	100%
L00196626	Oleg	Dvoinisiuk	100%
L00196692	Kristine	Kaulina	100%
L00196743	Brian	McNamara	100%
L00186189	Sergiy	Kochenko	100%
L00203052	*Peace*	*Agbonlahor Awana*	*0%*

2. Project Selection

- Selected Project: ATM Application Project, team's own project
- Language: Python
- Project Category: Web Service
- This project was selected by the team due to its familiarity with a real-world problem and practical user experience. It allows the team to apply key programming concepts in a realistic context.

- **Key Features**
  - Successful connection to API.
  - Successful deployment on such as Huroku or Render (optional).
  - Card reader, for reading a customer's card.
  - Pin pad, for entering a pin code and other information, such as payment / withdrawal amounts.

3. Initial Technical Design

- **Core Classes**
  1. Account - Stores details of the account holder and their balance on the account.
  2. ATM Card - Stores the card number, pin and a reference to the Account. An account holder may have more than 1 card linked to their account.
  3. Validator - Will be used to check the correct pin is used against the card number, the response generated will determine the next step.
  4. ValidationAttempt - Record all successful and unsuccessful attempts for security reporting (Reporting not required in this app, but the class will be used to monitor unsuccessful attempts).

5. Transaction - Record all withdrawal/lodgement transactions.
6. Encryption - Encrypt/Decrypt the messages between the "Terminal" and the "Server".
7. Terminal - Controls the interactions with the user, sends messages to the Server.
8. Server - Receives messages from the Terminals and handles appropriately.

- **Key Data Structures and Algorithms**
  - Main Data Structures: Lists and class objects (cardHolder) will be used to store and manage client data retrieved from Google Sheets.
  - Algorithm Implementation: Linear search (to find users), input validation (for PIN and amounts), and update algorithms (to modify balances in Google Sheets).

- **Testing Strategy**
  - Testing Approach: The project will utilize a combination of unit testing and manual testing. Unit tests will be developed following a structured approach to validate core functionalities and ensure early detection of defects.
  - The Most Challenging Parts to Test: The most challenging aspect will be writing unit tests for specific and complex test cases, particularly those involving interdependent modules or dynamic data conditions.
  - Ensuring 80%+ Test Coverage: To achieve 80% or higher test coverage, the coverage tool unittest will be used to measure and report on test completeness. Additional manual testing will be performed to assure 100% test coverage. Regular coverage reports will guide additional test case creation to improve under-tested areas.

4. Work Plan

- The team decided not to assign specific roles, allowing all members to participate in various aspects of the project and gain broader experience. The Kanban tasks will be created in GitHub, and each team member will assign tasks to others to ensure balanced workload distribution, and collaboration.
- Communication Plan: Weekly MS Teams meetings, with ongoing collaboration and information sharing via MS Teams and GitHub.
- Git Workflow Strategy: Main development branch and features.
- Development Tools and Environment: The project will be developed on Desktop Visual Studio and commit to the GitHub repository. Git Bash app will be used.

5. Risk Assessment

- **Potential Technical Challenges**
  - Varied Programming Experience: Team members have different levels of experience with programming languages such as Python and Java. Some have advanced coding knowledge, while others have limited or no prior experience. The choice of programming language for this project has been determined by considering both the team members' prior experience and the specific requirements of the course. This may delay the project's progress, as some team members may require additional time to become familiar with the chosen programming language.
  - Experience with GitHub & Visual Studio Environments: Team members have varying levels of experience using GitHub for version control and collaboration. This was addressed during the group's first meeting through a brief knowledge-sharing session, followed by additional self-study for less experienced members. The same approach was applied to Visual Studio, where a more experienced team member facilitated a walkthrough to help others become familiar with the development environment.
  - Engagement and Motivation: Keeping the project engaging for experienced members while manageable for beginners will be a challenge. Experienced members might find tasks too easy, while less experienced ones might feel overwhelmed and slow progress. To address this, tasks should be assigned based on skill level, making sure everyone remains fully engaged.

- **Unequal Team Contribution Handling**
  - Decision-Making in a Large Group: Making decisions on technical and design issues can take a lot of time. The team must establish a clear decision-making process and documentation. Every team member should have a chance to contribute their opinion before the decisions are finalized.
  - Clear Responsibilities: Clearly define responsibilities at the beginning of the project, considering each member's skill level, capacity, and the area of interest. Periodically review and adjust assignments to maintain a fair and balanced workload, considering everyone's private schedules.

- **Contingency Plan for Delays in Team Schedule**

  To avoid delays in the project, schedule the following actions should be taken:

- The project scope should be established early, outlining the project stages and tentative deadlines.
- A project management plan should be maintained, including clear agendas, to-do lists, and documented meeting notes and action items.
- Well-structured weekly meetings should be held to review progress, discuss next steps, and address any emerging issues.
- Open communication should be encouraged regarding delays, challenges, or personal matters so that tasks can be reprioritized or workloads adjusted to keep the project on track.