# Report 1: Project Proposal Report (Initiation Phase)

## Project Title

Coffee Shop Chain Management System

## Team Members and Roles

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Student ID | Student Name | Role |
|  | HE190006 | Nguyễn Trọng Hiệp | All |

## Objectives and Expected Outcomes

**Objectives:**

* Develop a robust Windows application using C# featuring a WinForms interface for primary user interaction and a Console App for administrative and testing purposes.
* Implement a secure, role-based access control system with two distinct roles:
  + Admin/Staff: Full permissions to create, read, update, delete, and search all data, including customers, menu items, staff profiles, and sales records.
  + Customer: Limited access to view the menu, manage loyalty program membership, and review personal purchase history.
* Design and build a comprehensive management interface within the WinForms application, organized into tabs for customers, menu items, staff, sales, and loyalty programs.
* Integrate functionality to generate a minimum of three key statistical reports (e.g., sales trends, most popular menu items, loyalty program engagement).
* Ensure secure logout functionality and implement role-based feature visibility through C# event handling.

**Expected Outcomes:**

* A fully functional and well-documented C# application (WinForms and Console App) that meets all specified requirements.
* A complete C# codebase adhering to object-oriented principles (encapsulation, inheritance, polymorphism).
* A suite of unit tests to ensure code quality and reliability.
* A final project that demonstrates proficiency in C# programming, database integration, and software design for the course assignment.

## Scope and Technical Requirements

**Scope:**

* Included: The project will cover the development of both a WinForms desktop application and a supplementary Console App. Core functionalities include user authentication, management of customers, menu items, staff, sales, and loyalty programs, and the generation of PDF reports. The system will connect to a Microsoft SQL Server database.
* Excluded: The application is designed for the Windows operating system only and will not support cross-platform deployment (macOS, Linux). The project will not include real-time payment gateway integration or advanced cloud-based services beyond what is required for the CI/CD pipeline.

**Technical Requirements**:

* Platform: .NET 8
* UI Framework: Windows Forms (WinForms)
* Command-Line Interface: Console App
* Programming Language: C#
* Database: Microsoft SQL Server
* Data Access: ADO.NET or Entity Framework
* IDE: Visual Studio 2022
* Version Control: Git and GitHub
* Report Generation: iTextSharp library for creating PDF reports.

## Initial Implementation Plan

|  |  |  |
| --- | --- | --- |
| Milestone | Timeline | Key Activities |
| First Presentation | Week of June 23-27, 2025 | - Present project overview, purpose, and objectives. - Outline implementation approach, advantages, and risks. - Submit initial report. |
| Second Presentation | Week 9 (August 18-22, 2025) | - Design database schema (ERD), use cases, and UI wireframes. - Implement initial C# code for WinForms and Console App. - Set up GitHub repository with CI/CD pipeline and unit tests. |
| Final Submission | Week 11 (September 6-12, 2025) | - Complete all application features for both WinForms and Console App. - Finalize all testing (unit, integration). - Generate sample reports and prepare final presentation/demo. |

## Resources and Tools

* Software & Frameworks: Visual Studio 2022, .NET 8, Windows Forms, Microsoft SQL Server, Entity Framework/ADO.NET.
* Version Control: A GitHub repository has been created for source code management.
* CI/CD: GitHub Actions will be configured for automated builds and execution of unit tests to ensure code quality throughout the development lifecycle.
* Testing: MSTest framework will be used for writing and running unit tests.
* Design: Microsoft Visio will be used for creating the ERD and UI wireframes.
* Documentation: Project documentation will be maintained in Markdown format on GitHub.

## Risk Assessment

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| C# Logic or Query Errors | Implement comprehensive unit and integration testing using a CI/CD pipeline to catch bugs early. Adopt a systematic debugging approach. |
| Security Vulnerabilities | Apply C# security best practices, including parameterized SQL queries to prevent SQL injection and proper input validation. |
| Scope Creep | Strictly adhere to the defined project scope. New features will be considered only after core requirements are met and if time permits. |
| Database Setup Complexity | Document the database setup process clearly. Use scripts to automate schema creation and data seeding where possible. |

## References (Optional)

* Microsoft Docs. (2025). C# Programming Guide. <https://docs.microsoft.com/en-us/dotnet/csharp/>
* Microsoft Docs. (2025). Windows Forms Overview. <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/>
* Microsoft Docs. (2025). ADO.NET Overview. <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/>
* iTextSharp Documentation. (2025). <https://itextpdf.com/>
* GitHub Guides. (2025). <https://guides.github.com/>