LAKEHEAD UNIVERSITY

FACULTY OF ENGINEERING

Software Project Management Plan

For the ChocAn Project

Instructor:	Dr. Ayman Diyab
Lab Instructor:	Mr. Mohammed AbuFoul

North Light Software

Full name	Student ID
Joshua Whitlock	1272349
Thanh Vu Nguyen	1244225
Kenneth Shahi	1261283

Thunder Bay, 2025

Contents

1 Overview

1.1 Project Summary

1.1.1 Purpose Scope and Objectives

The goal is to build ChocAn's Project, a software product that supports provider terminal and ChocAn database interaction. The product will let the client add, edit, and delete the needed records, run the required checks, and produce weekly reports and payment files.
Objectives:

- Deliver a working product that meets all stated requirements and ship the product to the client on time.

- Produce clear reports and correct totals.

1.1.2 Assumptions and Constraints

- We must meet the deadline and stay within the budget.

- The product must be reliable and easy to maintain.

- The design must allow new features later.

- The user interface must be simple.

1.1.3 Project Deliverables

- Working product.

- User guide.

- Technical notes and setup guide.

- Test results and acceptance record.

1.1.4 Schedule and Budget Summary

Workflows with time, people, and internal cost.

- Requirements workflow – 3 week two people $0.

- Analysis workflow – 1 week two people $0.

- Design workflow – 1 week two people $0.

- Implementation workflow – 1 week three people $0.

- Testing workflow – 1 week three people $0.

- Total 10 weeks and $0.

1.1.5 Evolution of the Plan

Any change to this plan must be approved by the engineering team, and recorded so the plan stays current.

2 Reference Materials

All work will follow the textbook standards for coding documents, testing, and reviews.

3   Definitions and Acronyms

List the terms used in this plan

- **ChocAn**: The client company.

- **Client**: the sponsor or user of the system.

- **DataCenter**: the main system that stores records and runs reports.

- **EFT**: electronic funds transfer file for provider payments.

- **UI**: user interface.

- **DB**: database.

4   Project Organization

4.1   External Interfaces

- **Client contact**: ChocAn Head Office/Dr. A. Diyab with questions.

4.2   Internal Structure

Team of three:

- **Kenneth**: software engineering student

- **Vu**: software engineering student

- **Josh**: software engineering student

4.3   Roles and Responsibilities

- Engineering communicates via e-mail if they have questions, keep records current on the git project, and escalate questions to Dr. Diyab via e-mail.

- Engineering Student A owns features [area A for example reports and operating data].

- Engineering Student B owns features [area B for example services and payments].

- Engineering Student C owns features [area C for example version tracking and mergers].

- Each member is responsible for the quality of their artifacts.

5   Managerial Process Plans

5.1   Start up Plan

5.1.1   Estimation Plan

Total time 7 weeks total internal cost $0 based on past class project deadline.

5.1.2   Staffing Plan

- Three engineering students are needed for all 10 weeks first 5 weeks as analysts and designers then as developers and testers.

5.1.3 Resource Acquisition Plan

All tools and machines are available. The product will be delivered as source code, and the client will be expected to compile it in their environment.

5.1.4 Project Staff Training Plan

The engineers will be responsible for **first** reviewing the textbook and **then** consulting AI with any follow-up questions about their roles.

5.2 Work Plan

5.2.1 Work Activities

- Week 1-3 gather and inspect requirements, meet with the client, define group setup, and confirm scope.

- Weeks 4 build analysis artifacts and inspect them, then get client approval.

- Weeks 5 build design artifacts and inspect them.

- Weeks 6-7 begin implementation by writing tests, then implement classes, integrate them, write docs, and do the final inspection.

5.2.2 Schedule Allocation

5.2.3 Resource Allocation

Each member works on assigned artifacts. Bi-weekly stand up to track progress and blockers. Bi-weekly client meeting (class) to confirm any scope changes.

5.2.4 Budget Allocation

Activity	Cost
Requirements	$0
Analysis	$0
Design	$0
Implementation	$0
Testing	$0
Total	$0

5.3 Control Plan

Any change that affects time or budget must be approved by the engineering team and recorded. Each person tests another person's work to keep quality.

5.4 Risk Management Plan

Main risks and how we track them

- No existing product to compare so we plan strong testing.

- The client has limited technical background so we keep screens simple and review often.

- Find any possible design fault so we inspect the design and write tests early.

- Possible machine or tool failure so we keep a spare machine or switch tools if needed.

- Performance risk is low due to small size but we will monitor response time.

5.5 Project Close out Plan

Give the client the product, user guide, test results, and acceptance note. Collect lessons learned.

6 Technical Process Plans

6.1 Process Model

We will use the Unified Process as in the textbook.

6.2 Methods Tools and Techniques

- **Language**: The primary project language will be Python, selected for its abundance of open source libraries.

- **Diagrams**: Diagrams will follow the standard for UML.

- **Source control**: Git and GitHub will be used for source control.

- **Issue tracking**: GitHub Issues will be used to flag and close issues.

6.3 Infrastructure Plan

Students will use a combination of their own laptops, their own operating systems, and collaborate through GitHub. Since Python was selected as the language and is interpreted, there will be no need for a compiler standard.

6.4 Product Acceptance Plan

The client accepts the product after:

- It is handed in on D2L and meets the questions asked of the students.

7 Supporting Process Plans

7.1 Configuration Management Plan

All files are stored in the git repository located here. We use branches for features, and pull requests for review.

7.2 Testing Plan

Unit tests for classes integration tests for flows and product tests for full scenarios. Each person first tests their code and then tests a teammate code. The git repository owner runs final integration tests.

7.3 Documentation Plan

We deliver the user setup steps, and final report. All docs follow the same format as Appendix F.

7.4 Quality Assurance Plan

- Peer reviews for all artifacts.

- When a team member finishes a feature they request another team member to test it via e-mail.

- The git owner does integration testing and product testing.

7.5 Reviews and Audits Plan

7.6 Problem Resolution Plan

Any major problem is recorded as a GitHub Issue at once. The github owner reads it, assigns it, sets a due date, and tracks to closure.

7.7 Subcontractor Management Plan

Not applicable.

7.8 Process Improvement Plan

We follow the team plan to improve repeatable practices and move toward higher maturity in the next term project. We will keep a record of our processes and lessons learned to iterate on our development process.

8 Additional Plans

8.1 Security

A password is needed to perform CRUD operations on the database. Providers login using their provider ID. All roles are limited to the needed actions.

8.2 Training

Training will be given by Vu at delivery. One day is expected. Questions for the first year are supported at no cost.

8.3 Maintenance

The team provides the software as-is, without warranty, with a full apache-2 license. Enhancements will use a separate agreement.