

**INTERNATIONAL SCHOOL**

**APPLICATION DEVELOPMENT PRACTICE**

**CMU-CS 246 NIS**

Version: 1.2

**PROJECT PLAN DOCUMENT**

**JAVA APPLICIATON CALCULATOR**

**Mentor: Nguyen Dang Quang Huy**

***Team Member*:**

**Nguyen Minh Nguyen**

**Nguyen Thi To Loan**

**Huynh Van Thien**

**Nguyen Van Tu**

**Ngo Xuan Bach**

**Nguyen Phuc Sang**

**Approved by Huy Nguyen Dang Quang**

**Project Plan Review Panel Representative:**

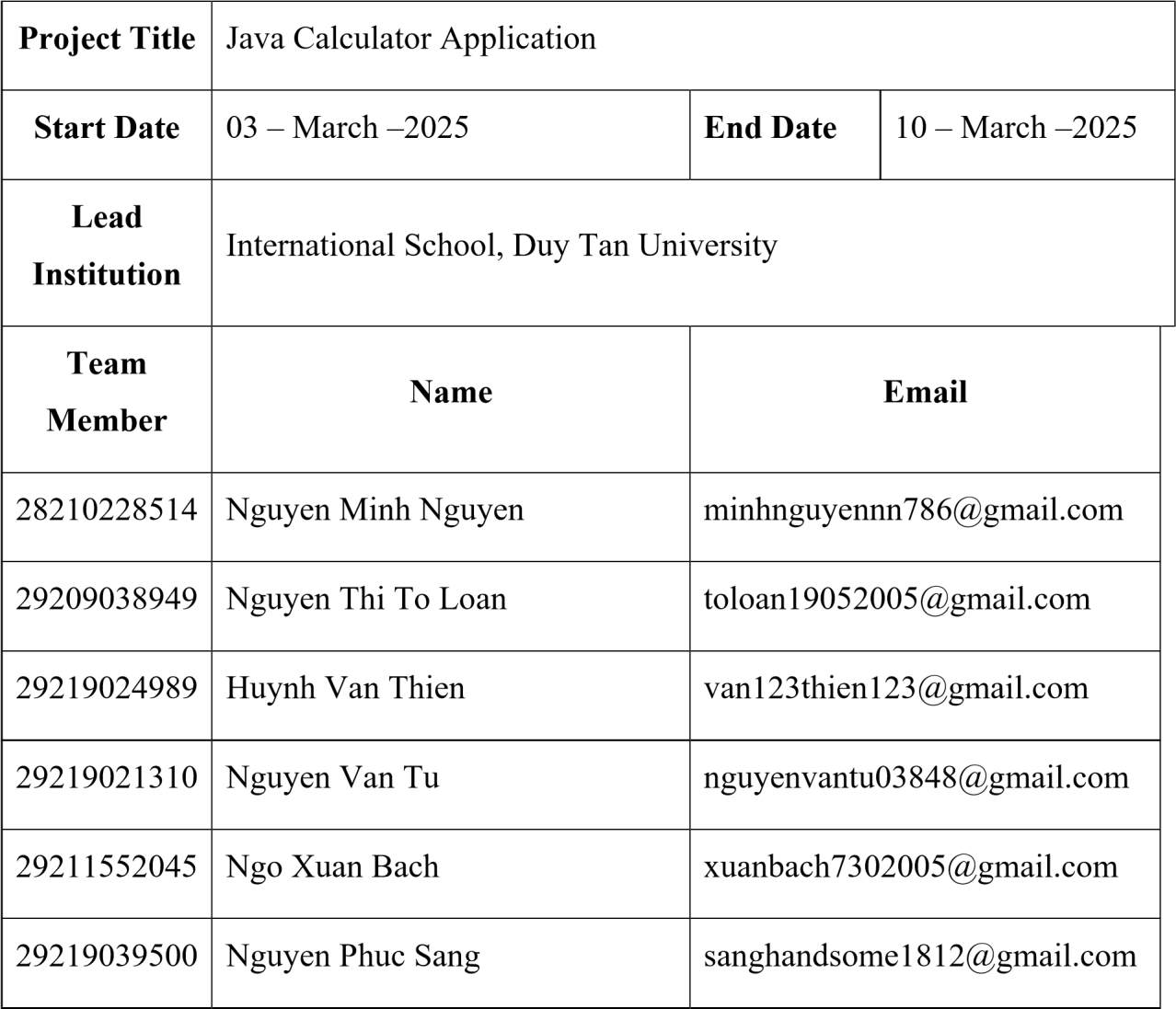
Name Signature Date

**Application Development Practice- Mentor:**

Name Signature Date

Da Nang, 03/2025

**PROJECT INFORMATION**



**PROJECT PLAN DOCUMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title** | Project Plan Document | | |
| **Author(s)** | CMU-CS 246 NIS | | |
| **Date** | March 3rd, 2025 | **File name:** | START-UP |
| **URL** | [https://drive.google.com/drive/folders/1oNEEhaTaH81316WZCgtx](https://drive.google.com/drive/folders/1oNEEhaTaH81316WZCgtxQOLMLz6XqoGi)  [QOLMLz6XqoGi](https://drive.google.com/drive/folders/1oNEEhaTaH81316WZCgtxQOLMLz6XqoGi) | | |
| **Access** | Project and CMU Program | | |

**SIGNATURE**

**Document Approvals:** The following signatures are required for approval of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mentor** | Nguyen Dang Quang Huy | **Signature:** |  |
| **Date:** |  |
| **Scrum**  **master** | Nguyen Minh Nguyen | **Signature:** |  |
| **Date:** |  |
| **Product**  **Owner** | Nguyen Thi To Loan | **Signature:** |  |
| **Date:** |  |
| **Team member(s)** | Huynh Van Thien  Nguyen Van Tu | **Signature:** |  |
| **Date:** |  |
| Ngo Xuan Bach Nguyen Phuc Sang | **Signature:** |  |
| **Date:** |  |

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Person(s)** | **Date** | **Description** |
| **1.0** | All members | March 3rd, 2025 | Draf |
| **1.1** | Nguyen Minh Nguyen | March 10th, 2025 | Final |
| **1.2** | Nguyen Minh Nguyen | April 16th ,2025 | Update version |

**PROJECT PLAN DOCUMENT**

Table of Contents

[1. PROJECT OVERVIEW 5](#_Toc13230)

[1.1 Project Description 5](#_Toc13231)

[1.2 Scope and Objectives 5](#_Toc13232)

[1.3 Assumptions and Constraints 5](#_Toc13233)

[2. SOFTWARE DEVELOPMENT PROCESS 6](#_Toc13234)

[2.1 Development Methodology 6](#_Toc13235)

[2.2 Tools and Technologies 6](#_Toc13236)

[3. SOFTWARE REQUIREMENTS SPECIFICATION (SRS) 6](#_Toc13237)

[3.1 Functional Requirements 6](#_Toc13238)

[3.2 Non-functional Requirements 6](#_Toc13239)

[3.3 Use Case Diagram 6](#_Toc13240)

[4. SYSTEM DESIGN 6](#_Toc13241)

[4.1 UI Mockups 6](#_Toc13242)

[4.2 System Architecture 6](#_Toc13243)

[6. TESTING PLAN 6](#_Toc13244)

[6.1 Unit Testing 6](#_Toc13245)

[6.2 Integration Testing 6](#_Toc13246)

[6.3 Test Cases 7](#_Toc13247)

[7. PROJECT SCHEDULE 7](#_Toc13248)

[8. TEAM ROLES & RESPONSIBILITIES 7](#_Toc13249)

[9. RISK MANAGEMENT 8](#_Toc13250)

[10. CONCLUSION 8](#_Toc13251)

# 1. PROJECT OVERVIEW

### 1.1 Project Description

This project involves the development of a simple calculator application using Java. The application will allow users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division, along with advanced functions like exponentiation, square root, and percentage calculation. The calculator will feature a user-friendly interface with real-time computation and error handling.

The enhanced version will also include scientific operations (factorial, logarithms, trigonometric functions), history storage, dark/light mode UI, keyboard shortcuts, and robust error handling.

### 1.2 Scope and Objectives

### The goal of this project is to create an advanced calculator application that supports basic and scientific operations, maintains a persistent history of calculations using file storage, and provides a customizable, user-friendly interface. The system also aims to support error detection and keyboard control for improved usability.

### 1.3 Assumptions and Constraints

* The application is developed for desktop use only.
* No persistent data storage; history is cleared when the application is closed.
* The UI should be simple and easy to navigate.

## 2. SOFTWARE DEVELOPMENT PROCESS

### 2.1 Development Methodology

We will follow an Agile methodology with an iterative approach. The project will be divided into multiple sprints, with continuous integration and testing.

### 2.2 Tools and Technologies

* Programming Language: Java
* IDE: Apache Netbeans/ Eclipse
* Version Control: GitHub
* Testing Framework: JUnit

# 3. SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

### 3.1 Functional Requirements

### Basic operations: Addition, Subtraction, Multiplication, Division

### Advanced operations: Exponentiation, Square Root, Percentage

### Scientific operations: Factorial (n!), Logarithm (log₁₀, ln), Trigonometric functions (sin, cos, tan, cot), Angle unit conversion (Degree ↔ Radian)

### History Management: Save and load history from a text or JSON file, search and delete specific entries

### UI Customization: Light/Dark mode, font and color settings

### Keyboard Shortcuts: Support full keyboard input for calculations

### Enhanced Control Features: CE (Clear Entry), Backspace and Forward button

### Clipboard Features: Copy/Paste result

### Expression Parsing: Evaluate complex expressions like “5 + (3 \* 2) - √9”

### Error Handling: Display descriptive error messages for invalid inputs (e.g., divide by zero, invalid syntax)

### 3.2 Non-functional Requirements

* Performance: The calculator should provide results instantly.
* Usability: The interface must be intuitive and easy to use.
* Maintainability: The code should be modular and well-documented.

### 3.3 Use Case Diagram

A Use Case Diagram will be included to illustrate the interactions between users and the system.

# 4. SYSTEM DESIGN

### 4.1 UI Mockups

The UI design will include buttons for each arithmetic operation and a display area for results.

### 4.2 System Architecture

The system will follow the Model-View-Controller (MVC) pattern for better maintainability.

# 6. TESTING PLAN

### 6.1 Unit Testing

JUnit will be used for unit testing to validate individual functions.

### 6.2 Integration Testing

Integration tests will verify the interaction between different modules of the application.

### Test Cases

* + - * + Test Case 1: 5 + 3 → Expected result: 8
        + Test Case 2: 10 / 2 → Expected result: 5
        + Test Case 3: Press 'C' → Expected result: Screen clears
        + Test Case 4: log₁₀(100) → Expected: 2
        + Test Case 5: sin(90°) → Expected: 1
        + Test Case 6: 5! → Expected: 120
        + Test Case 7: Switch to Dark Mode → Expected: UI background changes
        + Test Case 8: Enter invalid expression "5++2" → Expected: Error message
        + Test Case 9: Copy result of calculation → Expected: Value saved to clipboard

# 7. PROJECT SCHEDULE

The project will follow a timeline with milestones, including requirement gathering, design, development, testing, and deployment.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Timeframe** | **Tasks** | **Responsible**  **Members** |
| **1. Requirement**  **Gathering** | Day 1 | * Define functional &   nonfunctional requirements. - Write the Software  Requirements Specification (SRS).   * Create user stories and backlog. | Nguyen Minh  Nguyen ,Nguyen Thi  To Loan, Nguyen  Van Tu |
| **2. Design** | Day 2 | - Design UI Mockups. - Draw system diagrams (Use  Case, Activity Diagrams). -  Plan software architecture (MVC). | Huynh Van Thien  ,Ngo Xuan Bach |
| **3.Implementation** | Days 3-4 | - Develop the frontend interface (Java Swing/JavaFX). - Implement core calculator logic (addition, subtraction, multiplication, division, exponentiation, square root). - Integrate frontend with backend. | Nguyen Minh  Nguyen |
| **4. Testing** | Day 5 | - Write and execute test cases (JUnit). | Nguyen Phuc Sang |
|  |  | * Perform functional and   UI testing.   * Identify and fix bugs. |  |
| **5. Deployment &**  **Documentation** | Days 6-7 | * Prepare final reports and user documentation. * Finalize the project and conduct a final review. * Present the project. | All members |

# 8. TEAM ROLES & RESPONSIBILITIES

|  |  |  |
| --- | --- | --- |
| Member | Assigned Feature Area | Responsibilities |
| Nguyen Minh Nguyen | Control Functions | Implement CE (Clear Entry), Backspace, Forward functionality |
| Ngo Xuan Bach | Calculation History | Develop history saving/loading via JSON/Text, search and delete functionality |
| Huynh Van Thien | Additional Features | Implement Copy/Paste, complex expressions parsing (e.g., “5 + (3 \* 2) - √9”) |
| Nguyen Phuc Sang | Advanced Error Handling | Detect and display meaningful errors (e.g., “5++2”, “√-9”, divide by zero) |
| Nguyen Van Tu | User Interface | Add Dark/Light mode, font and color customization, keyboard shortcuts |
| Nguyen Thi To Loan | Scientific Operations | Implement factorial, log/ln, sin/cos/tan/cot, and degree-radian conversion |

# 

# 9. RISK MANAGEMENT

Potential risks include delays in implementation, bugs in code, and unclear requirements. Mitigation strategies include thorough testing, peer reviews, and regular team meetings.

# 10. CONCLUSION

This project aims to deliver a user-friendly and efficient calculator application with robust functionality and smooth user experience.