**Project Document: College Registration System**

Table of Contents

1. Introduction

1.1. Purpose of the Document

1.2. Project Overview

1.3. Scope

2. System Requirements

2.1. Functional Requirements

3. Architecture

3.1. High-Level Architecture

3.2. Class Diagram

4. User Interface

5. Technologies Used

6. Testing

6.1. Test Cases

6.2. Unit Testing

7. Conclusion

8. References

**1. Introduction**

**1.1. Purpose of the Document**

The purpose of this document is to provide an overview of the College Registration System project developed using C++. It outlines the system's requirements, architecture, features, user interface, testing procedures.

**1.2. Project Overview**

The College Registration System is standalone console application that allows users to perform various activities, such as student management, module management etc. The system aims to provide an event driven interface for student/admins to access the system.

**1.3. Scope**

The scope of the Student Registration System project includes the following functionalities:

- Add Student Record.

- Update Student Record.

- Display Student Record

- Add Modules.

- Display Modules Details

- Update Modules

**2. System Requirements**

**2.1. Functional Requirements**

1. Add Student Record: User should be able to create new student record.

2. Update Student Record: User should able to update existing user record.

3. Display Student Record: System should display all student record.

4. Add Module: User should able to add a new module.

5. Display Module Details: System should able to display all the modules.

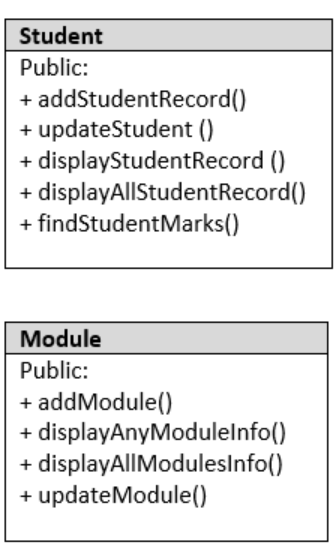
6. Update Modules: Users should be able update a given module.

**3. Architecture**

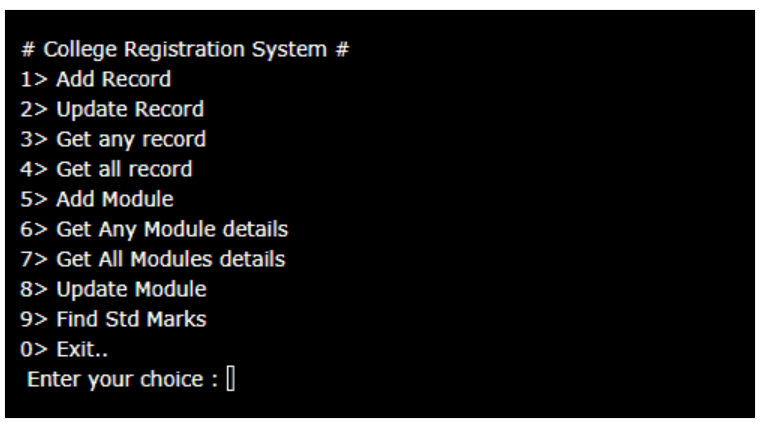
**3.1. High-Level Architecture**

The College Registration System will follow a standalone console-based event driven architecture. .

**3.2. Class Diagram**



**4. User Interface**



**5. Technologies Used**

C++, Oracle Database

**6. Testing**

**6.1. Test Cases**

\*(Provide a set of test cases that will be executed during the testing phase. Include positive and negative test scenarios.)\*

**6.2. Unit Testing**

For this project do manual testing.

**7. Conclusion**

Learner should able to understand use of arrays, loops, fileIO, database integration etc.

**8. References**

\*(Include any references to external resources, documentation, or libraries used during the project.)\*