**TABLE OF CONTENT**

1. Introduction

1.1 Project Details: Broad specifications of the work entrusted to you

1.2 Purpose

1.3 Scope

1.4 Objective

1.5 Technology and Literature Review

2. Project Management

2.1 Feasibility Study

2.1.1 Technical feasibility

2.1.2 Time Schedule feasibility

2.1.3 Operational feasibility

2.1.4 Implementation feasibility

2.2 Project Planning

2.2.1 Project development Approach and Justification

2.2.2 Project Plan

2.2.3 Role and Responsibilities

3. System Requirements Study

3.1 Study of Current System

3.2 Problems and Weakness of Current System

3.3 User Characteristics

3.4 Hardware and Software Requirements

3.5 Constraints

4. System Analysis

4.1 Requirements of New System (SRS)

4.1.1 Functional Requirements

4.1.2 Non-functional requirements

5. System Design

5.1 Use Case Diagram

5.2 Class Diagram

5.3 Sequence Diagram

5.4 Activity Diagram

6. Implementation Planning

6.1 Implementation Environment

6.2 Program Specification

7. Testing

7.1 Testing Plan

7.2 Testing Strategy

7.3 Testing Methods

7.4 Test Cases

8. User Manual

9. Limitation and Future Enhancement

10. Conclusion and Discussion

10.1 Conclusions

10.2 Discussion

10.2.1 Self Analysis of Project Viabilities

10.2.2 Problem Encountered and Possible Solutions

10.2.3 Summary of Project work

11. Bibliography

1.1 Project Details

The broad specification of work for a full-stack development project typically involves various tasks and responsibilities across the front-end and back-end development. Here's an outline of the work that may be entrusted for this project :

1. Project Planning and Requirement Analysis.
2. Develop responsive and user-friendly web pages using HTML, CSS, and JavaScript
3. Implement interactivity and dynamic content using front-end frameworks like React js.
4. Choose and implement an appropriate back-end technology stack (e.g., Node.js, Django, Flask, Ruby on Rails, etc.).
5. Design and implement the database schema based on the application's data model
6. Implement authentication and authorization mechanisms.
7. Conduct unit testing for both front-end and back-end components.
8. Optimize code and infrastructure for performance

1.2 Purpose