**Capstone Project and Research 2 Manuscript Document Format**

**1. Title Page**

* **Project Title**: The title of the capstone project.
* **Proponents**: Names of the students.
* **Degree Program**: Bachelor of Science in Information Technology.
* **Institution**: Eastern Visayas State University - Ormoc Campus.
* **Date of Submission**: Month and year of submission.

**2. Approval Sheet**

* **Signature of Adviser**: Sign-off from the project adviser.
* **Signature of Panel Members**: Sign-off from the defense panel members.
* **Signature of Department Head**: Approval from the department head.
* **Date of Approval**: Date when the project was approved.

**3. Acknowledgments**

* A section to acknowledge individuals, organizations, and institutions that provided support and guidance throughout the project.

**4. Abstract**

* **Summary of the Project**: A concise summary of the project’s objectives, methodology, key findings, and conclusions.
* **Keywords**: A list of keywords relevant to the project.

**5. Table of Contents**

* A detailed table of contents listing all sections, subsections, tables, and figures with their corresponding page numbers.

**6. List of Figures**

* A list of all figures used in the document, along with their titles and page numbers.

**7. List of Tables**

* A list of all tables used in the document, along with their titles and page numbers.

**8. Chapter 1: Introduction**

* **Background of the Study**: Overview of the problem and context in which the project is situated.
* **Problem Statement**: Clear and concise statement of the problem being addressed.
* **Objectives**: General and specific objectives of the project.
* **Significance of the Study**: The importance and contributions of the project to the field of Information Technology.
* **Scope and Limitations**: The boundaries of the project and any constraints.
* **Definition of Terms**: Key terms and concepts defined.

**9. Chapter 2: Review of Related Literature**

* **Related Works**: A detailed review of literature, software, and algorithms relevant to the project.
* **Theoretical Framework**: Theories and models that underpin the project.

**10. Chapter 3: Methodology**

* **Research Design**: The design of the research (qualitative, quantitative, or mixed methods).
* **System Design and Development**: Details on the design and development of the system, including tools and technologies used.
* **Data Collection and Analysis**: Methods used for collecting and analyzing data.
* **Implementation Plan**: Steps and timeline for implementing the project.
* **Testing and Evaluation**: Procedures for testing and evaluating the system.

**11. Chapter 4: Results and Discussion**

* **Presentation of Data**: Detailed presentation of the results obtained from the implementation.
* **Analysis of Results** **and Discussion:** In-depth analysis and interpretation of the data. Discussion on how the results relate to the project objectives and problem statement.

**12. Chapter 5: Conclusion and Recommendations**

* **Conclusion**: Summary of the findings and how they address the objectives of the study.
* **Recommendations**: Suggestions for further improvements or future research.

**13. References**

* **Bibliography**: List of all sources cited in the document, formatted according to the chosen citation style (APA, ).

**14. Appendices**

* **Appendix A**: Survey Questionnaires, Interview Guides, or other research instruments used.
* **Appendix B**: Screenshots or code snippets relevant to the project.
* **Appendix C**: Additional data or materials that support the findings.
* **Appendix E**: Curriculum Vitae of the Proponents.

**Grading System for Capstone Project and Research 2**

The grading system for Capstone Project and Research 2 is designed to evaluate the students' performance across various stages of project development, implementation, documentation, and presentation. The final grade will be a combination of the following components:

**1. Project Development (40%)**

* **System Design (10%)**: Evaluation of the design process, including architecture, flowcharts, and overall system planning.
* **Implementation and Coding (20%)**: Quality of code, adherence to best practices, and successful implementation of the system.
* **Testing and Debugging (10%)**: Effectiveness of the testing process, identification and resolution of bugs, and overall system stability.

**2. Project Documentation (20%)**

* **Technical Documentation (10%)**: Completeness, accuracy, and clarity of the technical report, including system specifications and design documentation.
* **User Manual (5%)**: Quality and usability of the user manual, providing clear instructions for end-users.
* **Final Report (5%)**: Overall presentation of the final report, including reflection on the project process, challenges faced, and lessons learned.

**3. Project Presentation and Defense (30%)**

* **Oral Presentation (10%)**: Clarity, organization, and delivery of the presentation, including the ability to effectively communicate project objectives, process, and outcomes.
* **Demonstration of System (10%)**: Functionality and performance of the system during the demonstration, including how well it meets the project objectives.
* **Panel Defense (10%)**: Ability to answer questions, defend the project decisions, and discuss the implications and future work.

**4. Teamwork and Professionalism (10%)**

* **Collaboration and Contribution (5%)**: Assessment of individual contributions to the team effort, as well as the effectiveness of teamwork.
* **Professionalism (5%)**: Professional behavior throughout the project, including meeting deadlines, communication with the adviser and panel, and adherence to ethical standards.

**Final Grade Computation:**

* **Project Development:** 40%
* **Project Documentation:** 20%
* **Project Presentation and Defense:** 30%
* **Teamwork and Professionalism:** 10%

**Passing Rate:** The passing rate for this course is 75%. Students must achieve a minimum of 75% in each component to pass the course.

**Grade Scale:**

* **95 - 100:** Outstanding
* **90 - 94:** Excellent
* **85 - 89:** Very Good
* **80 - 84:** Good
* **75 - 79:** Satisfactory
* **Below 75:** Needs Improvement / Failing

This grading system ensures that students are assessed comprehensively, taking into account not just the final product, but the entire process of project development, from inception to final presentation.

**Guidelines for Writing the "System Design and Development" Section in the Capstone Project and Research 2 Manuscript**

**Introductory Paragraph**

The "System Design and Development" section is a crucial part of the Capstone Project and Research 2 manuscript. This section provides a comprehensive overview of how the system was designed, developed, and implemented. It should include detailed descriptions of the architecture, tools, technologies, methodologies, and processes used throughout the project. The purpose is to give readers a clear understanding of the technical aspects of the project and how the system meets the objectives outlined in the proposal.

**1. System Architecture and Design**

* **Overview of the System Architecture**: Start by providing a high-level overview of the system's architecture. This should include a description of the overall system structure, including the different components, modules, and how they interact with each other.
  + **Example**: "The system architecture is divided into three main layers: the presentation layer, the business logic layer, and the data access layer. Each layer has specific responsibilities and interacts with the other layers to perform the system's functions."
* **Design Diagrams**: Include design diagrams such as flowcharts, data flow diagrams (DFD), Unified Modeling Language (UML) diagrams, and entity-relationship (ER) diagrams. These diagrams should visually represent the system's structure and flow of information.
  + **Example**: "Figure 1 shows the UML diagram for the system, depicting the main classes and their relationships."
* **Database Design**: Discuss the database design in detail, including the structure of tables, relationships, primary and foreign keys, and any normalization techniques applied.
  + **Example**: "The database is structured into five main tables: Users, Transactions, RFID Logs, Student Information, and Entry\_Exit Logs. Each table has been normalized to the third normal form to eliminate redundancy and ensure data integrity."

**2. Development Process**

* **Development Methodology**: Explain the development methodology followed, such as Agile, Waterfall, Scrum, or any other relevant approach. Describe why this methodology was chosen and how it was applied during the development process.
  + **Example**: "The Agile methodology was chosen for its flexibility and iterative approach, allowing for continuous feedback and improvement throughout the development cycle."
* **Tools and Technologies Used**: Provide a detailed list of the tools, technologies, and programming languages used during the development. Explain the rationale behind choosing these tools and how they were used in the project.
  + **Example**: "The system was developed using Python for backend development due to its robust libraries and frameworks. The frontend was built using HTML, CSS, and JavaScript, while MySQL was used for database management."
* **Coding Standards and Best Practices**: Discuss the coding standards and best practices followed during development. Mention any version control systems (e.g., Git) used to manage the codebase.
  + **Example**: "The team adhered to PEP 8 coding standards for Python to ensure code readability and maintainability. Git was used for version control, allowing multiple developers to collaborate efficiently."

**3. Challenges and Solutions**

* **Challenges Faced During Development**: Discuss any significant challenges encountered during the design and development phases. These could include technical difficulties, resource limitations, or any other obstacles.
  + **Example**: "One of the major challenges faced was optimizing the RFID scanning process to handle large volumes of students during peak hours. This was addressed by implementing a multi-threaded approach to improve system responsiveness."
* **Solutions Implemented**: Explain the solutions or workarounds that were implemented to overcome these challenges.
  + **Example**: "To address the challenge of handling large volumes of data, the team implemented a load-balancing algorithm that distributes the scanning process across multiple servers."

**4. Summary of the Design and Development**:

Conclude this section by summarizing the key aspects of the system's design and development. Highlight the major accomplishments and how the system meets the objectives outlined in the proposal.

* + **Example**: "The system's design and development have been guided by the project's objectives, resulting in a robust and efficient Monitoring and Verification System for Student Entry and Exit using RFID technology. The system is scalable, secure, and meets the specific needs of the institution."

**Guidelines for Writing the "Implementation Plan"**

The "Implementation Plan" section of your Capstone Project and Research 2 manuscript is crucial as it outlines how the project will be developed and deployed. It provides a roadmap of the steps and strategies to be followed for successful implementation.

**1. Introductory Paragraph**

* **Purpose**: Start with a brief introduction that explains the purpose of the Implementation Plan. Mention how it will guide the project from development to deployment.
* **Overview**: Provide an overview of what will be covered in this section, including key phases of implementation, timelines, and responsibilities.

**2. Project Phases**

* **Planning and Preparation**: Describe the initial phase where you will finalize requirements, gather resources, and set up the development environment.
  + **Requirements Finalization**: Confirm all the functional and non-functional requirements.
  + **Resource Allocation**: Identify hardware, software, and human resources needed.
  + **Environment Setup**: Detail the setup of development and testing environments.
* **Development**: Outline the coding and system development phase.
  + **Module Development**: Break down the system into modules, detailing the development of each.
  + **Version Control**: Mention the use of version control systems (e.g., Git) for managing code.
  + **Integration**: Explain how the modules will be integrated to form the complete system.
* **Testing and Quality Assurance**: Discuss the testing strategies that will ensure the system functions as intended.
  + **Unit Testing**: Testing of individual components or modules.
  + **Integration Testing**: Testing how the modules work together.
  + **User Acceptance Testing (UAT)**: Explain how end-users will test the system to ensure it meets their requirements.
  + **Bug Fixing and Refinement**: Describe the process for addressing any issues found during testing.
* **Deployment**: Detail how the system will be deployed to the live environment.
  + **Deployment Strategy**: Choose between direct deployment, phased rollout, or parallel deployment.
  + **Data Migration**: If applicable, describe how existing data will be transferred to the new system.
  + **Go-Live**: Define the steps leading up to the system going live.
* **Post-Deployment Support and Maintenance**: Describe plans for system support, monitoring, and future updates.
  + **Support**: Outline how support will be provided to users after deployment.
  + **Monitoring**: Describe the tools and processes for monitoring the system's performance.
  + **Maintenance**: Detail how the system will be maintained and updated as needed.

**3. Timeline and Milestones**

* **Gantt Chart**: Provide a visual timeline using a Gantt chart that shows the start and end dates for each phase.
* **Key Milestones**: Identify critical milestones, such as the completion of the development phase, testing phase, and go-live date.

**4. Resource Allocation**

* **Human Resources**: List the team members involved in each phase and their responsibilities.
* **Budget**: If applicable, provide a budget overview for each phase, including costs for resources, tools, and other expenses.

**5. Risk Management**

* **Risk Identification**: Identify potential risks that could impact the implementation.
* **Mitigation Strategies**: Describe the strategies that will be used to mitigate these risks.
* **Contingency Plans**: Outline contingency plans in case of unexpected challenges.

**6. Communication Plan**

* **Stakeholder Communication**: Explain how progress will be communicated to stakeholders.
* **Reporting**: Define the frequency and format of progress reports.
* **Feedback Loops**: Describe how feedback will be collected and integrated into the project.

**7. Summary**

* **Summary**: Summarize the key points of the Implementation Plan.
* **Commitment to Success**: Reinforce the commitment to following the plan to ensure the successful completion of the project.

**Tips for Writing the Implementation Plan**

* **Be Specific**: Provide detailed descriptions for each phase and task.
* **Use Clear Language**: Avoid jargon and ensure the plan is understandable by all stakeholders.
* **Stay Realistic**: Set realistic timelines and resource allocations.
* **Be Flexible**: While the plan should be detailed, allow some flexibility for adjustments during the implementation.

**Example Section Heading Structure**

1. Introductory Paragraph
2. Project Phases
   * Planning and Preparation
   * Development
   * Testing and Quality Assurance
   * Deployment
   * Post-Deployment Support and Maintenance
3. Timeline and Milestones
4. Resource Allocation
5. Risk Management
6. Communication Plan
7. Summary

### **Guidelines for Writing the "Testing and Evaluation" Section**

The "Testing and Evaluation" section of the Capstone Project and Research 2 manuscript is critical as it demonstrates how the developed system was tested to ensure it meets the requirements and performs as expected. This section should cover the strategies, methodologies, and criteria used for testing, as well as the outcomes of the evaluation process. Below are the guidelines for writing this section:

#### 1. **Introductory Paragraph**

* **Purpose**: Begin by explaining the purpose of the Testing and Evaluation section. Mention that this section will describe how the system was tested and evaluated to ensure it meets the project objectives and requirements.
* **Overview**: Provide a brief overview of the testing strategies, types of tests conducted, and the evaluation process.

#### 2. **Testing Strategies**

* **Types of Testing**: Describe the different types of testing used in the project. Common types include:
  + **Unit Testing**: Testing individual components or modules to ensure they function correctly.
  + **Integration Testing**: Ensuring that different modules or components work together as expected.
  + **System Testing**: Testing the entire system as a whole to ensure it meets the specified requirements.
  + **User Acceptance Testing (UAT)**: Involving end-users to validate that the system meets their needs and expectations.
  + **Performance Testing**: Evaluating the system's performance under various conditions (e.g., load testing, stress testing).
  + **Security Testing**: Ensuring the system is secure from vulnerabilities and threats.
* **Test Environment**: Describe the environment in which the testing was conducted, including hardware, software, and network configurations.
* **Test Data**: Explain the data used for testing, including how it was generated or selected, and how it represents real-world scenarios.

#### 3. **Test Plan**

* **Test Cases**: Provide details on the test cases developed for the project. A test case should include:
  + **Test Case ID**: A unique identifier for each test case.
  + **Description**: A brief description of what the test case is designed to verify.
  + **Input Data**: The data inputs used in the test case.
  + **Expected Results**: The expected outcomes if the system functions correctly.
  + **Actual Results**: The actual outcomes observed during testing.
  + **Pass/Fail Criteria**: The criteria used to determine whether the test case passes or fails.
* **Test Execution**: Describe the process for executing the test cases, including the sequence of tests and any tools or automation used.
* **Test Schedule**: Provide a timeline or schedule for when different tests were conducted.

#### 4. **Evaluation Criteria**

* **Functional Requirements**: List the functional requirements the system was tested against and describe how these were evaluated.
* **Non-Functional Requirements**: Discuss the evaluation of non-functional requirements such as performance, usability, reliability, and security.
* **Quality Assurance**: Explain the quality assurance measures taken to ensure the system meets the required standards.

#### 5. **Results of Testing**

* **Test Results Summary**: Provide a summary of the test results, including the number of test cases executed, passed, and failed.
* **Defects and Issues**: List any defects or issues identified during testing, along with their severity and the actions taken to resolve them.
* **Bug Tracking**: Mention any bug tracking tools or processes used to manage and resolve issues.
* **Lessons Learned**: Discuss any insights or lessons learned from the testing process that could be applied to future projects.

#### 6. **Evaluation of System Performance**

* **Performance Metrics**: Define and discuss the key performance metrics used to evaluate the system (e.g., response time, throughput, resource usage).
* **Benchmarking**: If applicable, compare the system's performance against industry standards or similar systems.
* **User Feedback**: Summarize feedback from users involved in User Acceptance Testing (UAT) and how it influenced the final system.

#### 7. **Summary**

* **Summary of Findings**: Summarize the key findings from the testing and evaluation process, including the overall success of the tests.
* **Implications for Final Deployment**: Discuss how the testing results will influence the final deployment of the system, including any last-minute changes or optimizations.

### **Tips for Writing the Testing and Evaluation Section**

* **Be Detailed**: Provide specific details for each testing phase, including test cases, data, and outcomes.
* **Use Tables and Charts**: Utilize tables, charts, and graphs to present test results and performance metrics clearly.
* **Stay Objective**: Present the findings in an objective manner, including both successes and areas that required improvement.
* **Document Changes**: If the testing led to changes in the system, document these changes and their impact.

### **Example Section Heading Structure**

1. Introductory Paragraph
2. Testing Strategies
   * Types of Testing
   * Test Environment
   * Test Data
3. Test Plan
   * Test Cases
   * Test Execution
   * Test Schedule
4. Evaluation Criteria
   * Functional Requirements
   * Non-Functional Requirements
   * Quality Assurance
5. Results of Testing
   * Test Results Summary
   * Defects and Issues
   * Lessons Learned
6. Evaluation of System Performance
   * Performance Metrics
   * Benchmarking
   * User Feedback
7. Summary