THE STATE UNIVERSITY OF ZANZIBAR

SCHOOL OF NATURAL AND SOCIAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

FINAL YEAR PROJECT REPORT FORMAT – Internet of Things (IoT) Project

Cover page

i. Declaration
ii. Abstract – a short summary stating the nature and scope of the work
iii. TABLE OF CONTENT
iv. TABLE OF FIGURES
v. ABRIVIATIONS

1. CHEPTER 1: Introduction

1.1. DESCRIPTION OF THE PROJECT AND BACKGROUND

Provide a short and clear description and intension of the project you are working on. For example, if you are working on a title called "Online Admission for SUZA", then you write about Online Admission in general. Also write about the organization for which you develop the system. Based on the previous title, this is about SUZA.

1.2. Problem statement –

Provide brief investigation of the system under consideration and gives a clear picture of what actually the physical system is?

BRIEF PROBLEM STATEMENT
Brief analysis or summary of the problems identified relating to the project or issue to be addressed by the project in not more than three paragraphs. Be sure to go and point out the problem explicitly. Also discuss about the effects of the problems on the company's operations

1.3. Problems Solution and the scope

Explain the scenarios of how your proposed system will be used and explain how this solution is better than the current practice. Describe the envisioned benefits of your proposed solution.

Write the scope of your project. To what extent your system solves the problem of the organization.
Generally, the services, the deliverability you have to accomplish in the given time

1.4. Objectives –

What you want to achieve at the end of the project. This is written as general /main objective. An example of statements in this part could be:

"The main objective of this project is"

There may also be specific objectives to accomplish the aim (main objective) of the project specific objective that help to achieve the general objective.
Identify the specific objectives:

a. The objective(s) should be precise, clear, Measurable, Relevant, achievable, and well defined.
b. The list of objectives should be presented in point form.
c. Consult supervisor for advice.

NB: [Four specific objectives are more than enough!]

1.5. **PROJECT BACKGROUND AND MOTIVATION**

Motivation:

Why the project is important / worth doing. Specify the significance of your project. What are effects of the project on the company or other company's in general. What is its contribution to the project area?

Background: What past works by others have been done on the problem?
To understand what your project is about, you should do a small survey of any related ideas and projects. which have been done so far which are related to your project. In essence, the ones that you have studied and hence lead to the identification of the problem/knowledge gap that you are trying to address. You should comment on existing works related to your problem.

1.6. Feasibility study report – operational, economic, legal and technical feasibility

Write about the feasibility of your project in terms of economic benefit, technical knowledge required to implement the system, and the time available for the project.

2. CHAPTER 2: Methodology

2.1. Requirement gathering methods

Discuss how you gather user requirements (interview, observation, document analysis, etc.)

2.2. Software development life cycle model(SDLC)

Describe what software development process style you have used in pursuing different phases of the software development life cycle. (E.g. Agile, Waterfall, Incremental, etc.). and justify why you decide to use that SDLC model.

3. CHAPTER 3: Requirements Analysis and Modeling

3.1 Requirement determination

3.1.1 Existing System

3.1.1.1 Existing System Description

The existing system description describes the current system of the organization as it is. This could be describing the activities they perform, how they handle information, and the drawbacks of the system.

3.1.1.2 Business Rules

Write the rules used by the organization currently. In the online banking case, this could be the interest rate allowed for saving account, the maximum amount of money that someone can withdraw at a time, interest rate for loan, etc. Generally, they are the rules by which the business is governed.

3.1.2 Proposed system

3.1.3 Functional requirements

Describe the functional requirements of the system. What functionalities does the system have? Then proceed to show it using preferred modelling techniques under the next section of requirement structuring

3.1.4 Non – functional requirements

Describe the non-functional requirements of the system like security, performance, reliability, etc.

3.2 Requirement Structuring

3.2.1 Process Modelling (if applicable)

If the project also involves the development of a database system for management, remote device control, and data visualization, the following sections detail its requirement modelling:

For OBJECT ORIENTED APPROACH

3.2.1.1 Use case diagram

Show the functionality of your system using use case diagram and how the actors interact with the system. Also show use case reusability by including <<include>>, <<extend>>, and <<inherit>> relationships between use cases.

3.2.1.2 Use Case Documentation

This is step by step description of the actions performed by each use case. It should contain preconditions, post conditions, main course of action, and alternate course of action.

3.2.1.3 Sequence diagram

Sequence diagrams should be drawn for each use case to show how different objects interact with each other to achieve the functionality of the use case.

For STRUCTURED APPROACH

3.2.1.1 Data Flow Diagram(DFD) to illustrate how current system is implemented

3.2.1.2 Data Flow Diagram (DFD to illustrate how proposed system is implemented

3.2.2 Data Modelling

For OBJECT ORIENTED APPROACH

3.2.2.1 Conceptual modelling: Class diagram

Create a class diagram that will be the building block the system you will develop. Class diagrams should show the objects the system is comprised of and how they are interrelated.

3.2.2.2 Entity relationship diagram

For STRUCTURED APPROACH

3.2.2.1 Entity relationship diagram

4. CHAPTER 4: System Design

4.1. Architectural design

4.1.1. High-Level Architecture Diagram

Describe the overall system architecture. This should include:

- Components: List and describe the various hardware and software components used in the project (e.g., sensors, actuators, microcontrollers, communication protocols).
- Connections: Explain how these components are connected and interact with each other within the IoT system.
- Cloud or Edge: Mention whether data processing and storage are done in the cloud or at the edge.

4.1.2. Hardware Prototype Design

4.1.2.1. Hardware Components

Outline the hardware components that will be used in the project, including: Sensors, Actuators, Microcontrollers/Processors, Communication Modules (Wi-Fi, Bluetooth, etc.), Power Supply. Describe each being used in terms of its specifications, its purpose, its pinouts and how it will be connected/configured.

4.1.2.2. Schematic Circuit Diagram

Create a detailed schematic diagram using standard electronic symbols to represent all hardware components and their connections. Include power supplies, resistors, capacitors, transistors (if applicable), and any other necessary components. Clearly label all components and their values (e.g., resistor resistances, capacitor capacitances). Annotate the diagram to explain signal flow and component interactions.

4.1.2.3. Flowchart diagram for Hardware Prototype firmware.

Create a flowchart diagram that visually represents the logic and execution flow of the microcontroller firmware. Use standard flowchart symbols (e.g., start/end boxes, decision diamonds, process rectangles) to depict the following aspects:

- Initialization: Power-up sequence, sensor configuration, communication module setup.
- Sensor Data Acquisition: Process for reading sensor data, including timing and data formatting.
- Data Processing (Optional): Any preliminary processing or filtering of sensor data on the microcontroller
- Decision Making (if applicable): Logic for making decisions based on sensor data (e.g., triggering actuators, sending data to remote server).
- Actuator Control (if applicable): Operations for controlling actuators based on sensor data or user commands.
- Communication: Sending data to a remote server or receiving control commands (if applicable).
- Error Handling: Strategies for handling potential errors (e.g., sensor malfunctions, communication issues).

4.1.2.4. Programming Description Diagram for Hardware Prototype firmware

- Create a PDL that details the pseudocode representation of the firmware logic.
- Use a structured approach, breaking down the firmware into modular functions or subroutines.
- Clearly define the purpose, inputs, and outputs of each function.
- Use keywords and control flow statements (e.g., if/else, loops) to represent the logic within each function.
- Maintain consistency in formatting and indentation for readability

4.2. Database-Centric System Design

If the project also involves the development of a database system for management, remote device control, and data visualization, the following sections detail its design:

4.2.1. Database Design

4.2.1.1. Relational Model

4.2.1.2. Data Description

4.2.1.3. Data Dictionaries

4.2.2. User Interface Design

4.2.2.1. Forms and Reports

4.2.2.2. Interface design sample

4.2.3. Access control and security

Within the system, different actors are granted access to specific functionality and data. Define the access controls for your system, ensuring security and proper management of user privileges.

4.3. Simulation of Prototype Design

- Describe the simulation process used to test the functionality and performance of the prototype design:
- Simulation Software: Specify the software used for simulation (e.g., Simulink, Proteus).
- Simulation Setup: Explain the setup of the simulation environment, including input parameters and expected outputs.
- Results Analysis: Discuss the results obtained from the simulation, highlighting any deviations from expected behavior and adjustments mad

4.4. Machine Learning Model Development (if applicable)

If the project involves machine learning:

Deployment Strategy

- Data source and Preparation: Describe the sources of training data how training data was prepared.
- Model Training: outline what Type of Model (e.g., Regression, Classification) you have used. Explain the process of training the machine learning model.
- Model Integration: How the trained model is integrated into the IoT system

4.5. IoT Cloud Storage (if applicable)

Data Selection for Cloud Storage:

- Identify the specific sensor data or processed information that will be uploaded to the cloud for storage and analysis.
- Consider factors such as data value for insights, storage requirements, and bandwidth limitations (if applicable).
- Explain the rationale behind the chosen data for cloud storage.

Cloud Storage Platform:

- If using a cloud storage platform (e.g., AWS S3, Microsoft Azure Blob Storage), specify the chosen platform and its justification.
- Briefly mention any relevant features or services offered by the platform that benefit the project (e.g., scalability, security).

Data Transmission Protocol:

- Define the communication protocol used to send data from the device to the cloud storage platform.
- Common options include:
  - ❖ HTTP: Versatile protocol suitable for various data formats, but may have higher overhead for frequent transmissions.
  - ❖ MQTT: Lightweight messaging protocol

5. CHAPTER 5: Implementation and Testing

5.1. Implementation

5.1.1. Code Structure and Organization

- Describe the structure of the code you have written, including the major components or modules, the functions or classes, and how they interact with each other. Provide a brief overview of the programming languages, frameworks, and libraries used.

5.1.2. Hardware Setup

- Outline the physical setup of the IoT system, including the arrangement of sensors, actuators, microcontrollers, and other hardware components. Include details such as wiring diagrams, placement of components, and any considerations made for real-world deployment (e.g., weatherproofing, power management).

5.1.3. Software Implementation

- Explain the process of developing the software that controls your IoT system. This should include:

- Firmware Development: Describe how the firmware was programmed into the microcontroller, highlighting key routines for sensor data collection, actuator control, and communication with the cloud.
- User Interface Implementation: If your project includes a user interface (UI), describe how it was developed, including the tools used and the main features provided (e.g., dashboards, control panels).

- Integration: Detail how different parts of the system (hardware, software, cloud services) were integrated to work together. Discuss any challenges faced during the integration process and how they were resolved.

5.2. Testing

5.2.1. Test Plan

- Describe the overall plan for testing the system, including the types of tests conducted (e.g., unit tests, integration tests, system tests), the testing environment, and the criteria used to determine if the system functions correctly.

5.2.2. Test Cases

Provide detailed descriptions of the test cases used, including:

- Objective: The purpose of each test.
- Inputs: The specific inputs or conditions for the test.
- Expected Outputs: What you expect the system to do in response.
- Actual Results: The observed behavior of the system when the test was run.
- Pass/Fail Criteria: How you determined whether the test was successful.

5.2.3. Test Results

- Summarize the results of the tests conducted. Highlight any issues found during testing, how they were resolved, and any remaining limitations or challenges that were not addressed.

5.2.4. Performance Evaluation

- If applicable, evaluate the performance of your system. This could include:

- Speed: How quickly the system responds to inputs or processes data.
- Accuracy: The precision of sensor readings or the correctness of actions taken.
- Scalability: How well the system can handle increasing loads, such as more devices or larger amounts of data.
- Reliability: How consistently the system performs under different conditions.

6. CHAPTER 6: Conclusion, Challenges, and Recommendations

6.1. Conclusion

6.1.1. Summary of Work Done

- Provide a concise summary of the entire project, highlighting the main objectives, the approach taken, and the outcomes achieved. Discuss the success of the project in meeting its goals and solving the identified problem.

6.1.2. Key Findings

- Summarize the most important findings from your project. This could include insights gained from the implementation and testing phases, unexpected results, or notable achievements.

6.1.3. Contributions to Knowledge/Industry

- Discuss the contributions your project has made to the field of IoT or to the specific industry or problem domain you were addressing. Mention how your work adds to existing knowledge or provides a solution that could be beneficial in real-world applications.

6.2. Challenges

6.2.1. Technical Challenges

- Describe the technical challenges encountered during the project, such as issues with hardware compatibility, software bugs, or difficulties in integrating different system components. Explain how these challenges were addressed or, if not resolved, how they impacted the project.

6.2.2. Operational Challenges

- Discuss any operational challenges, such as time constraints, resource limitations, or difficulties in acquiring necessary materials or data. Describe the impact these challenges had on the project timeline or scope.

6.3. Recommendations

6.3.1. For Future Work

- Suggest areas where the project could be expanded or improved in future work. This could include adding new features, improving system performance, or exploring different use cases or environments.

6.3.2. For Industry/Academia

- Provide recommendations based on your findings that could be useful for industry practitioners or academic researchers. This might include best practices, potential areas of research, or considerations for deploying IoT systems in the field.

6.3.3. Lessons Learned

- Reflect on the lessons learned throughout the project. Discuss what went well, what could have been done differently, and how these lessons could be applied to future projects.

References