Software Design DESCRIPTION

Web-Based System for Nano Material Repository

By: Muhamad Effendi bin Md Nor Kamal

Matric No.: S61973

Supervised by: Ts. Mohammad Aizat bin Basir

CSF4998 FINAL YEAR PROJECT I

FACULTY OF COMPUTER SCIENCE AND MATHEMATIC
UNIVERSITI MALAYSIA TERENGGANU

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Purpose

The purpose of this Software Design Description (SDD) is to provide complete specifications and design details for the development of a Web-Based System for Nano Material System. This SDD is intended to translate the collected requirements into a specific system architecture, components, interfaces, and data design that can guide the implementation phase of this project. The main functionality covered in this document includes allowing students to search and submit experimental data from journal. It also includes the ability for employees and stakeholders to view and access inventory. The primary audience for this SDD is the software development team, including the lead developer and any other programmers working on building this system who will Use it to guide coding and deployment. Additionally, this document serves as a reference for verification by the project supervisor to ensure that the design meets the stated requirements and scope. It can also help all software testing resources understand system design and develop effective test plans Thomas et al. (2013).

1.2 Scope

The software being specified in this SDD is the Nano Material Research Repository System. This web-based system will allow students to search for and submit experimental data and results from journal articles related to nano materials research. The core capabilities of the system include:

- User accounts and role-based access for students and staff members to submit or view data.
- Explain what the software product(s) will, and, if necessary, will not do
- Searchable database of nano materials journals and articles.
- Submission forms and templates for entering experimental data such as morphology, XRD, I-V characteristics, and UV-Vis Spectra.
- Database storage of submitted data linked to relevant journal articles.
- Dynamic views and reports summarizing submitted research data.
- Administrative functions for user management and data moderation.

  The focus is on capturing experimental data and measurements that can be numerically analyzed. Detailed experimental procedures or methodology will not be handled. The system will not store full journal articles, only the key results data.

  The goal is to create a centralized open repository of nano material research results that can enable further analysis and knowledge discovery. It aims to benefit students by giving them access to a wider set of structured experimental data for their own research.

1.3 Definition, Acronyms and Abbreviations

- Nano - Referring to the nano centimeter scale.
- XRD - X-Ray Diffraction, an experimental technique used to analyze the crystal structure of materials.
- I-V Characteristic - Current-Voltage characterization of nano structures.
- UV-Vis - Ultraviolet-Visible spectroscopy, used to characterize optical properties.
- SDD - Software Design Description.
- UI - User Interface.
- API - Application Programming Interface.
- SQL - Structured Query Language.

1.4 Overview

The remainder of this Software Design Description provides a comprehensive technical specification for the Nano Material Repository system. The SDD is organized into the following sections:

- **System Architecture -** Provides an overview of the structural design of the system, including the network topology, hardware requirements, and system components.
- **Software Architecture -** Details the software architecture patterns used, the breakdown of functionality into modules, and the interfaces between modules.
- Data Design - Specifies the logical and physical data models, database schemas, and data storage approaches.
- **Interface Design -** Provides specifics on the various user interfaces, APIs, and integration points with external systems.
- Security and Access Control Design - Describes methods for authentication, authorization, encryption, and other security mechanisms.
- **Release and Transition Plan -** Outlines the phases for development, testing, and deployment.

The document aims to provide the engineering team with the comprehensive details required to implement the system that meets the requirements outlined in the requirements analysis Hendren et al. (2015).

CHAPTER 2

SYSTEM ARCHITECTURAL DESIGN

2.1 Architectural Design

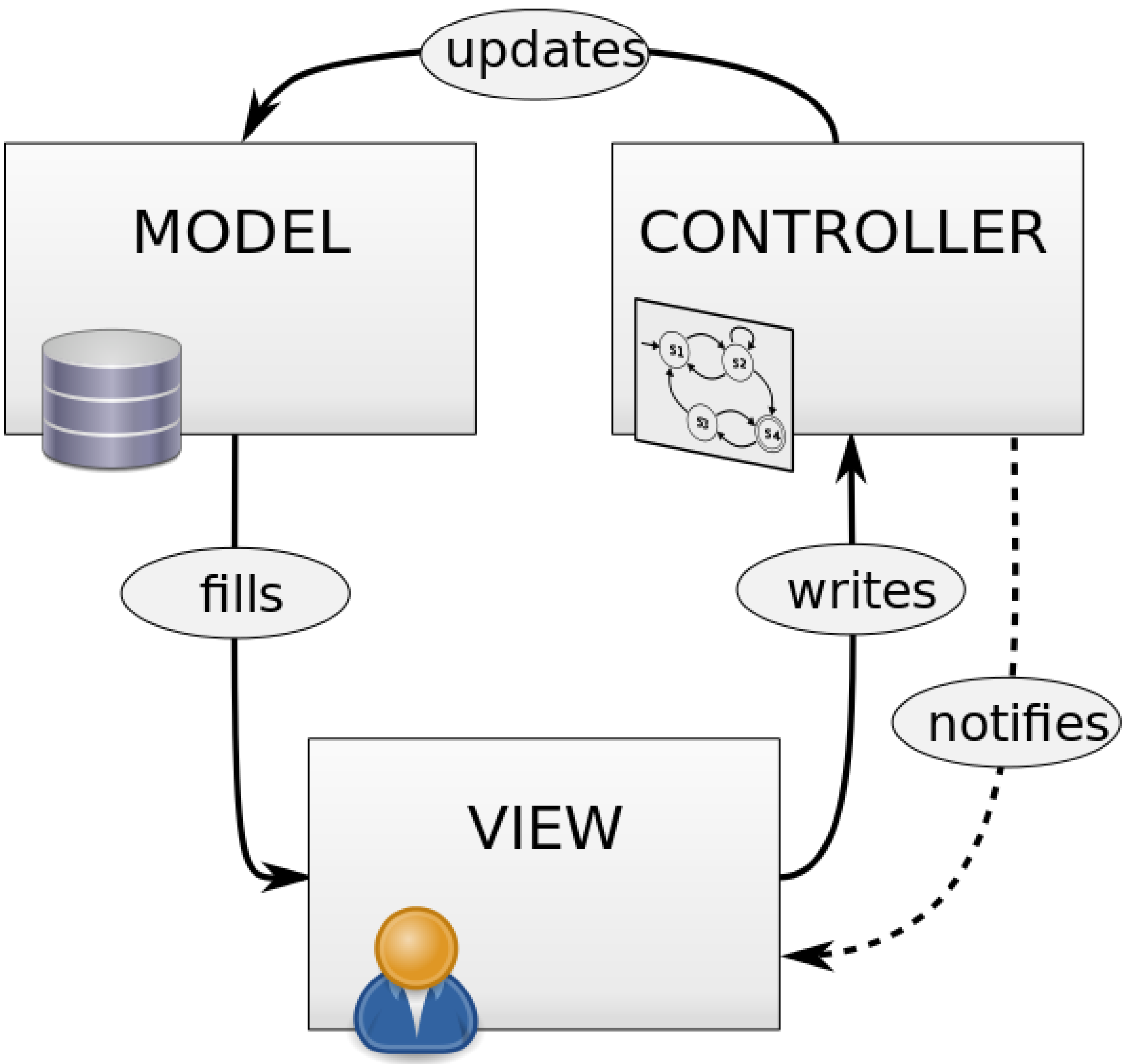


Figure 2.1: MVC-Model

- **Model:** The Model represents the data structures. Typically, the model classes will contain functions that help retrieve, insert, and update information in your database. In the context of your project, the Model would interact with the database to store and fetch information about the nano materials and the results of the experiments. The Model is independent of the user interface Pop and Altar (2014).
- **View:** The View renders presentation of the model in a particular format for the end user, typically a web page. The View generates an output presentation to the user based on changes in the Model. In the project, the View would be responsible for taking the data from the Model and displaying it to the user in a comprehensible format.
- **Controller:** The Controller responds to the user input and performs interactions on the data model objects. It receives user input and decides what to do with it. It's kind of a bridge between the Model and the View. For the project, the Controller would handle the user inputs, process them, interact with the Model to carry out the necessary actions, and then render the appropriate View Deacon (2009).

In this architecture, the Controller is the main point of interaction for the users. It communicates with the Model to store and retrieve data, and with the View to render the appropriate user interface. The Model interacts with the Database to fetch and store data Mufid et al. (2019).

2.2 Design Rationale

The Model-View-Controller (MVC) architecture was chosen for your nano material repository system due to its clear separation of concerns, which facilitates easier maintenance and understanding of the codebase. Each component in MVC has a distinct responsibility, making it easier to update, add, or modify features. MVC is a widely used and understood architecture, which aids other developers in understanding the structure of your system. This is particularly useful in a university setting, where your project may be passed on to future students for further development or learning. If you're working in a team, MVC allows multiple developers to work simultaneously on different components of the same application without affecting each other. While MVC can sometimes result in unnecessary complexity as the application scales, this is unlikely to be a significant issue given the scope of your project. Other architectures like Service-Oriented Architecture (SOA) could have been considered. These architectures could provide greater modularity and scalability. However, they also require more careful planning, coordination, and are typically used for larger, enterprise-level applications. Given the scope and requirements of your project, MVC is a more suitable choice Falessi et al. (2013).

CHAPTER 3

DATABASE DESIGN

3.1 Normalization

3.1.1 Zero Normal Form: 0NF

studentID	materialType	testType	concerntration	temperature	testData
testType	materialID	testinfoID			

Figure 3.1: Zero Normal Form

.

3.1.2 One Normal Form: 1NF

studentID	materialType	testType	concerntration	temperature	testData

Figure 3.2: FIrst Normalized Form

3.1.3 Second Normal Form: 2NF

testID	studentID	materialType	testType

testID	concerntration	temperature	testData

Figure 3.3: Second Normalized Form

3.1.4 Third Normal Form: 3NF

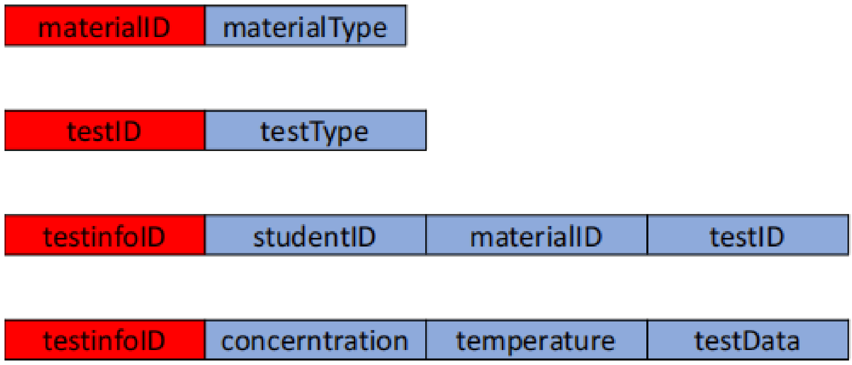


Figure 3.4: Third Normalized Form

3.2 Entity Relational Diagram

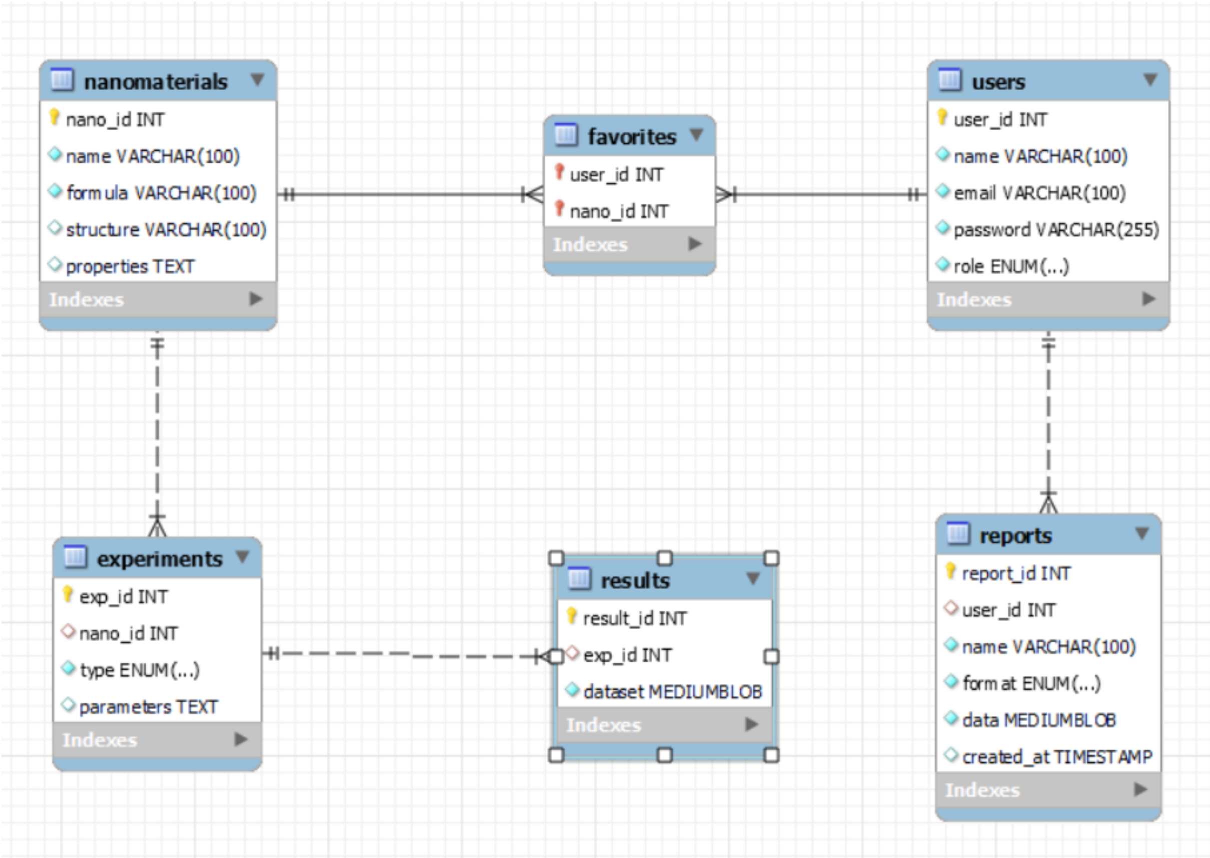


Figure 3.5: Entity Relational Diagram

3.3 Data Dictionary

Entity	Type	Description
Concerntration	VARCHAR(255)	The concentration level of the material used in the test
materialID	INT	A unique identifier for each type of material
materialType	VARCHAR(255)	The type of material being tested
studentID	INT	A unique identifier for each student
Temperature	VARCHAR(255)	The temperature level at which the test is conducted
testData	VARCHAR(255)	The data or results obtained from the test
testDataID	INT	A unique identifier for each test data.
testID	INT	A unique identifier for each type of test
testInfoID	INT	A unique identifier for each test instance
testType	VARCHAR(255)	The type of test or experiment being conducted

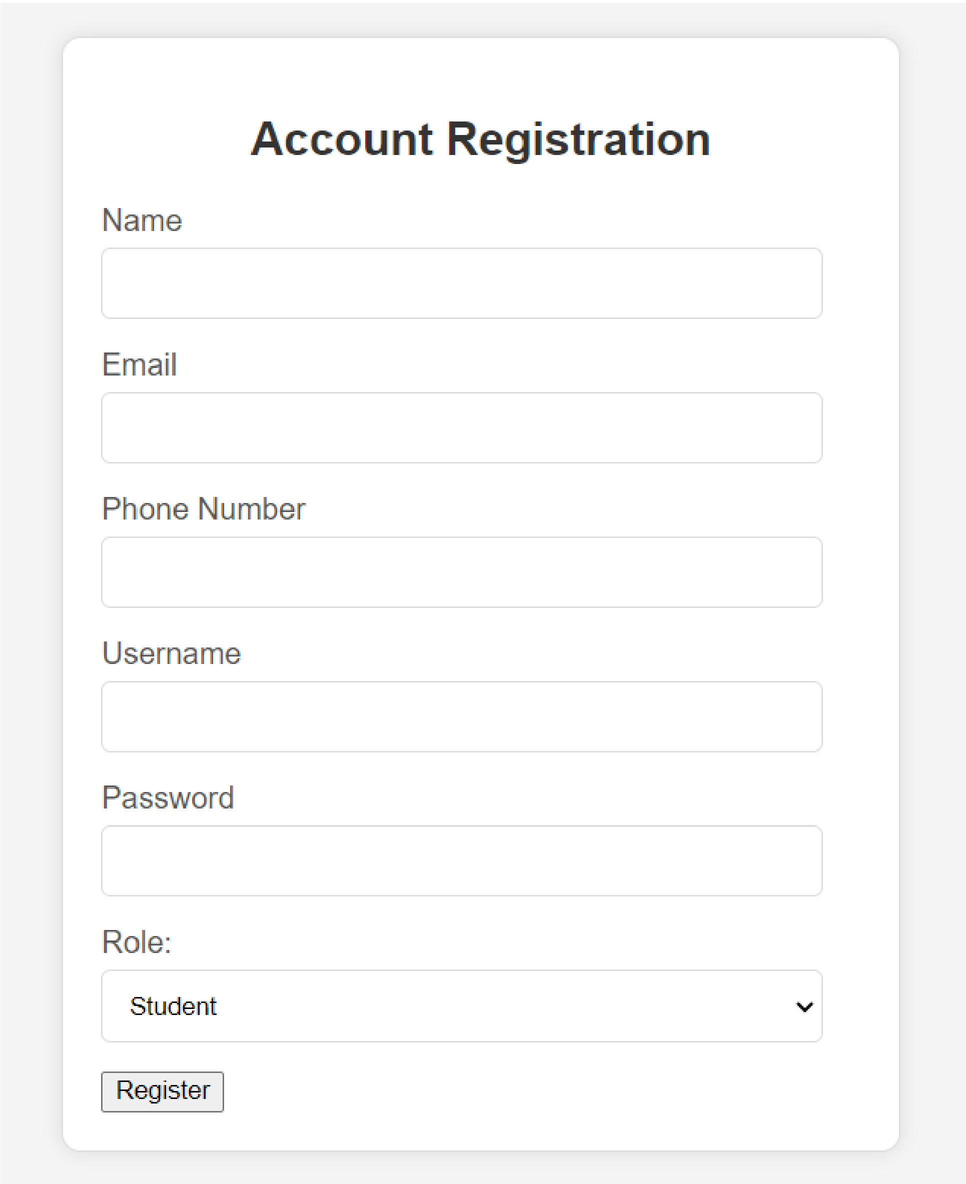
Figure 3.6: Data Dictionary

.

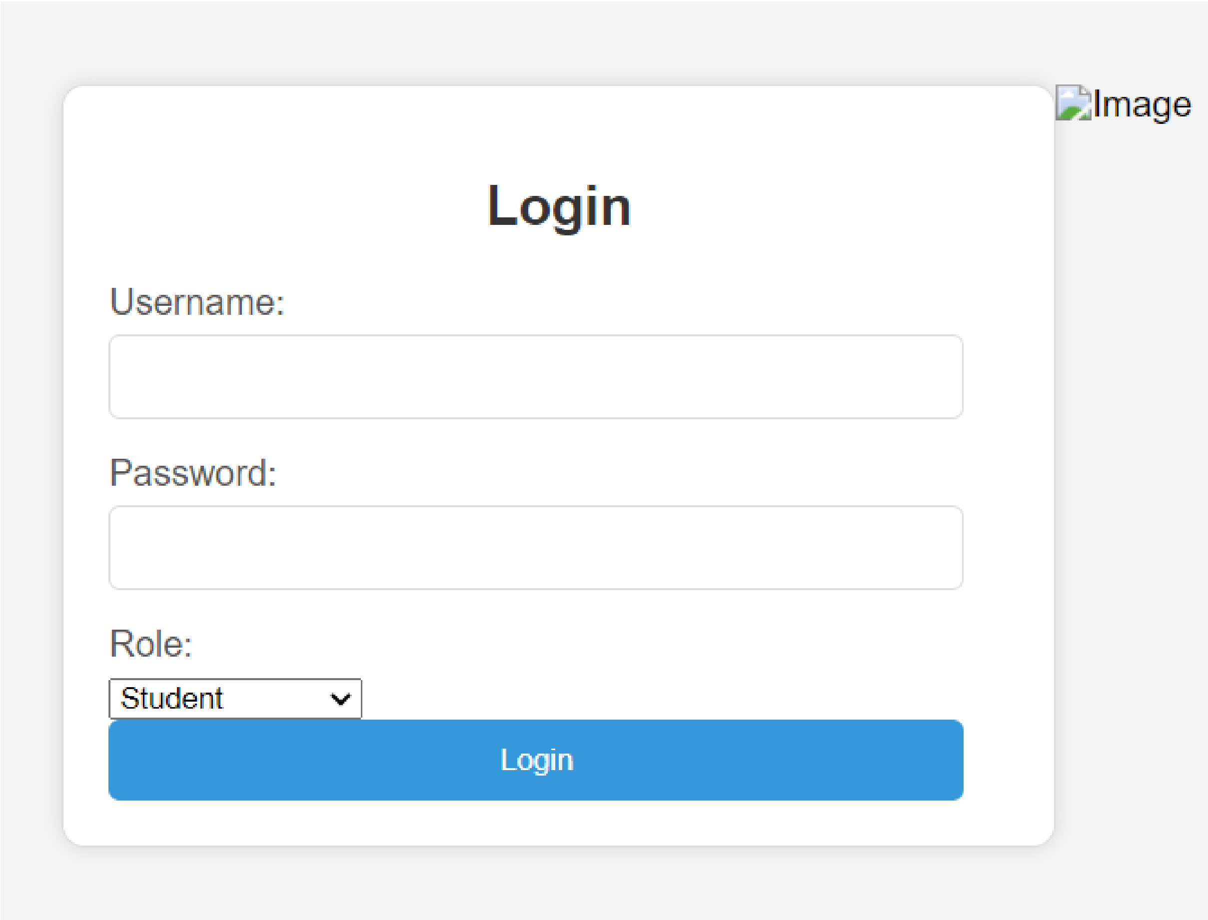
CHAPTER 4

UI/UX DESIGN

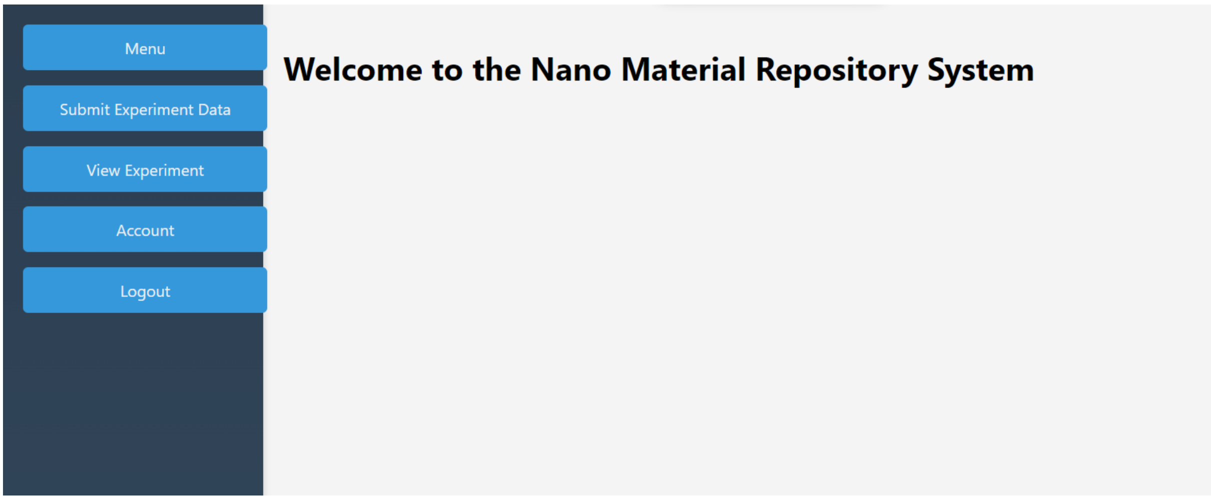
4.1 Registration Page



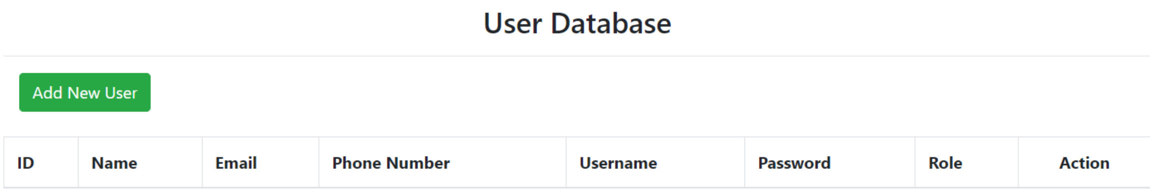
4.2 Login Page



4.3 Homepage Page



4.4 Account Database List



CHAPTER 5

CONCLUSION

The Software Design Document (SDD) for the web-based system for a nano material repository provides a comprehensive architectural overview of the system. It outlines the system's database design, which is normalized to the third normal form (3NF) to ensure data integrity and efficiency. The database includes tables for Students, Materials, Tests, Test Information, and Test Data, with relationships appropriately defined among them. The SDD also includes a data dictionary that provides a clear understanding of the major data entities, their types, and what they represent. The system is designed to handle various tests on different materials under varying conditions, making it a robust and flexible solution for managing and analyzing nano material data. The SDD serves as a blueprint for the development, deployment, and maintenance of the web-based system. It ensures that the system is built consistently and can be easily understood by all stakeholders, including developers, project managers, and end-users. The system, once implemented, promises to be a valuable tool for students and researchers working with nano materials.

REFERENCES

Deacon, J. (2009). Model-view-controller (mvc) architecture. *Online][Citado em: 10 de março de 2006.] http://www. jdl. co. uk/briefings/MVC. pdf*, 28.

Falessi, D., Briand, L. C., Cantone, G., Capilla, R., and Kruchten, P. (2013). The value of design rationale information. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 22(3):1–32.

Hendren, C. O., Powers, C. M., Hoover, M. D., and Harper, S. L. (2015). The nanomaterial data curation initiative: A collaborative approach to assessing, evaluating, and advancing the state of the field. *Beilstein journal of nanotechnology*, 6(1):1752–1762.

Mufid, M. R., Basofi, A., Al Rasyid, M. U. H., Rochimansyah, I. F., et al. (2019). Design an mvc model using python for flask framework development. In *2019 International Electronics Symposium (IES)*, pages 214–219. IEEE.

Pop, D.-P. and Altar, A. (2014). Designing an mvc model for rapid web application development. *Procedia Engineering*, 69:1172–1179.

Thomas, D. G., Gaheen, S., Harper, S. L., Fritts, M., Klaessig, F., Hahn-Dantona, E., Paik, D., Pan, S., Stafford, G. A., Freund, E. T., et al. (2013). Isa-tab-nano: a specification for sharing nanomaterial research data in spreadsheet-based format. *BMC biotechnology*, 13(1):1–15.