**Software Requirements Specification for Demir Export Maden Stok Takip**

**Version 1.0**

**Prepared by**

İlayda Gündüz

Merve Nur Erdem

Deniz Alagöz

**For Client:** Demir Export

**22.07.2025**

Table of Contents

[1. Introduction 3](#_Toc204941288)

[1.1 Purpose 3](#_Toc204941289)

[1.2 Scope 3](#_Toc204941290)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc204941291)

[1.4 References 4](#_Toc204941292)

[1.5 Overview 4](#_Toc204941293)

[2. General Description 4](#_Toc204941294)

[2.1 Product Perspective 4](#_Toc204941295)

[2.2 Product Functions 5](#_Toc204941296)

[2.3 User Characteristics 5](#_Toc204941297)

[2.4 General Constraints 5](#_Toc204941298)

[2.5 Assumptions and Dependencies 6](#_Toc204941299)

[3. Specific Requirements 6](#_Toc204941300)

[3.1 External Interface Requirements 6](#_Toc204941301)

[3.1.1 User Interfaces 6](#_Toc204941302)

[3.2 Functional Requirements 10](#_Toc204941303)

[3.2.1 Login Functionality 11](#_Toc204941304)

[3.2.2 Change Password Functionality 12](#_Toc204941305)

[3.2.3 Sample Adding and Listing 14](#_Toc204941306)

[3.2.4 Material Adding and Listing 17](#_Toc204941307)

[3.2.5 Stock Entry and Exit 20](#_Toc204941308)

[3.2.6 Admin Operations 23](#_Toc204941309)

[4. Other Nonfunctional Requirements 25](#_Toc204941310)

[4.1 Performance Requirements 25](#_Toc204941311)

[4.2 Security Requirements 25](#_Toc204941312)

[4.3 Software Quality Attributes 25](#_Toc204941313)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| SRS | 22.07.2025 | First Version Created | 1.0 |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

This document specifies the requirements of the **“Demir Export Maden Stok Takip”** software project, which aims to monitor and manage the stocks of mine samples stored on shelves and in refrigerator. The system will allow users to manage stock digitally instead of using paper-based records.

## 1.2 Scope

The **Demir Export Maden Stok Takip** project is a desktop-based application designed to manage and track the stock of mine samples stored in shelves and refrigerator. The system will allow users to perform stock input and output operations, monitor material quantities, and generate reports. The system includes a login mechanism to ensure that only authorized users can access stock-related features. The main goal is to eliminate manual, paper-based tracking and provide a more efficient and reliable digital solution. The application will be used by mining engineers and authorized staff within the company. This system does not include online access, mobile functionality, or integration with external databases.

## 1.3 Definitions, Acronyms, and Abbreviations

**UI:** User Interface: Visual elements that users interact with in the application.

**Admin:** A system user with privileges to manage users, view all data, and perform operations like editing and deleting users.

**Sample:** A physical or digital item whose stock is tracked in the system.

**Material:** Other name for consumable products.

**CRUD:** Create, Read, Update, Delete operations performed on data such as users, samples, and materials.

**Entity:** A distinct object in the system, such as a user, sample, or material.

**Validation:** A process of checking data input for correctness and completeness.

**DEStok Takip:** The original Turkish name of the project. "DE" stands for Demir Export, the name of the company, and "Stok Takip" translates to "Stock Tracking" in English.

## 1.4 References

"IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984 , vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.

## 1.5 Overview

This Software Requirements Specification (SRS) document defines all functional and non-functional requirements of the **Demir Export Maden Stok Takip** project, as well as its constraints, assumptions, and scope.

* **Section 1** provides the purpose, scope, definitions, and references for the project.
* **Section 2** outlines the overall description of the system, including user roles, product perspective, and system environment.
* **Section 3** presents detailed functional requirements that are essential for the software developer to implement the system.
* **Section 4** specifies non-functional requirements such as performance, usability, security, and technological constraints.

This document serves as a complete guide for both the development team and stakeholders to ensure a shared understanding of the system to be developed.

# 2. General Description

## 2.1 Product Perspective

**Demir Export Maden Stok Takip** system is a standalone, desktop-based stock tracing application developed in Java. It is designed to be installed and used on local machines without requiring internet access or any third-party services. The system is intended for internal use within the organization and will be distributed free of charge.

Users will log in to the system using credentials initially created and assigned by the system administrator. After their first login, users will have the ability to change their own passwords through the application. All authentication and user management are handled entirely within the system—no external or third-party tools will be integrated.

This version of the application is a **demo release** and may be extended or modified in the future based on user feedback and evolving requirements. However, potential future features and improvements are **not within the scope** of this current version.

## 2.2 Product Functions

This product will have the functions below:

1. Ensures the proper organization of laboratory shelves and cold storage to maintain a tidy and efficient workspace.
2. Allows users to register incoming and outgoing samples into the system with all relevant information. Ensures that each sample is stored according to laboratory standards.
3. Tracks the stock levels of materials and equipment in the laboratory. Users can view current inventory status.
4. Allows tracking of the importance level before disposal decisions.
5. Supports weight-based categorization and placement of materials in appropriate storage areas in the warehouse to ensure safety and efficient space usage.
6. Monitors all movements of inventory items, including additions, removals, and transfers between storage areas. Logs each movement for audit purposes.
7. Regular users can log in to the system and perform tasks such as registering sample input and output, monitoring stock levels, filtering items, searching for items, and organizing shelf and cold storage samples.
8. Admins are responsible for managing user accounts. This includes creating users, assigning initial passwords, and monitoring user activities in the system.

## 2.3 User Characteristics

Demir Export Maden Stok Takip will be used by Demir Export mining engineers without any product membership. Users don’t have to have advanced computer skills.

## 2.4 General Constraints

1. The application is developed in Java and JavaFX technologies.
2. The application will initially support only the Turkish language.
3. The application must provide a simple, user-friendly graphical user interface (GUI), designed for non-technical users.
4. As a demo version, the application is not designed to support large-scale enterprise data loads or concurrent multi-user access.
5. The application should be able to function without internet access, except for future update features.
6. There are two main roles: Admin and Standard User. Only admins can manage user accounts and monitor user activity.
7. Passwords are set by the admin and can be changed by users later. No third-party authentication or encryption services are used.
8. The system must be operable on computers with at least 4 GB RAM and a dual-core processor.

## 2.5 Assumptions and Dependencies

1. It is assumed that users can operate desktop software and perform basic tasks such as logging in, data entry, and navigation within the application.
2. It is assumed that the system will run on secure, organization-managed devices where external access is restricted.
3. The application assumes that only admins will assign initial passwords and monitor user activity.
4. All operations are assumed to be performed locally.
5. It is assumed that users will enter accurate and complete information according to organizational procedures.
6. It is assumed that the number of concurrent users and the volume of data will remain within demo-scale limits.
7. The system depends on a consistent and stable local database (MySQL) for all data operations.
8. The user interface design assumes standard screen sizes for optimal display and usability.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

##### 3.1.1.1 Login Page

This is the entry page to the system. It allows users to log in using their credentials.

ekran görüntüsü, dikdörtgen, tasarım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.2 Admin Page

Accessible only to admin users. It allows management of system users.

metin, yazılım, bilgisayar simgesi, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.3 Main Page

The central hub for regular users to access available operations.

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.4 User Profile Page

Allows users to view and update their personal profile details.

metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.5 Sample Add/List Page

Allows users to add new samples and view existing ones.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.6 Sample Entry/Exit Page

Used to record the in/out movements of samples.

metin, yazılım, bilgisayar simgesi, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

##### 3.1.1.7 Material Add/List Page

Lets users manage materials by adding new entries or listing existing ones.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

## 3.2 Functional Requirements

This section describes the functional requirements of the *Demir Export Maden Stok Takip* system. The figure below illustrates the Use Case Diagram, which represents the interactions between users and the system functionalities.

metin, ekran görüntüsü, diyagram, tasarım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

The next figure presents the Data Model, which outlines the relationships between core entities in the system database.metin, doküman, belge, ekran görüntüsü, diyagram içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

### 3.2.1 Login Functionality

##### 3.2.1.1 Description and Priority

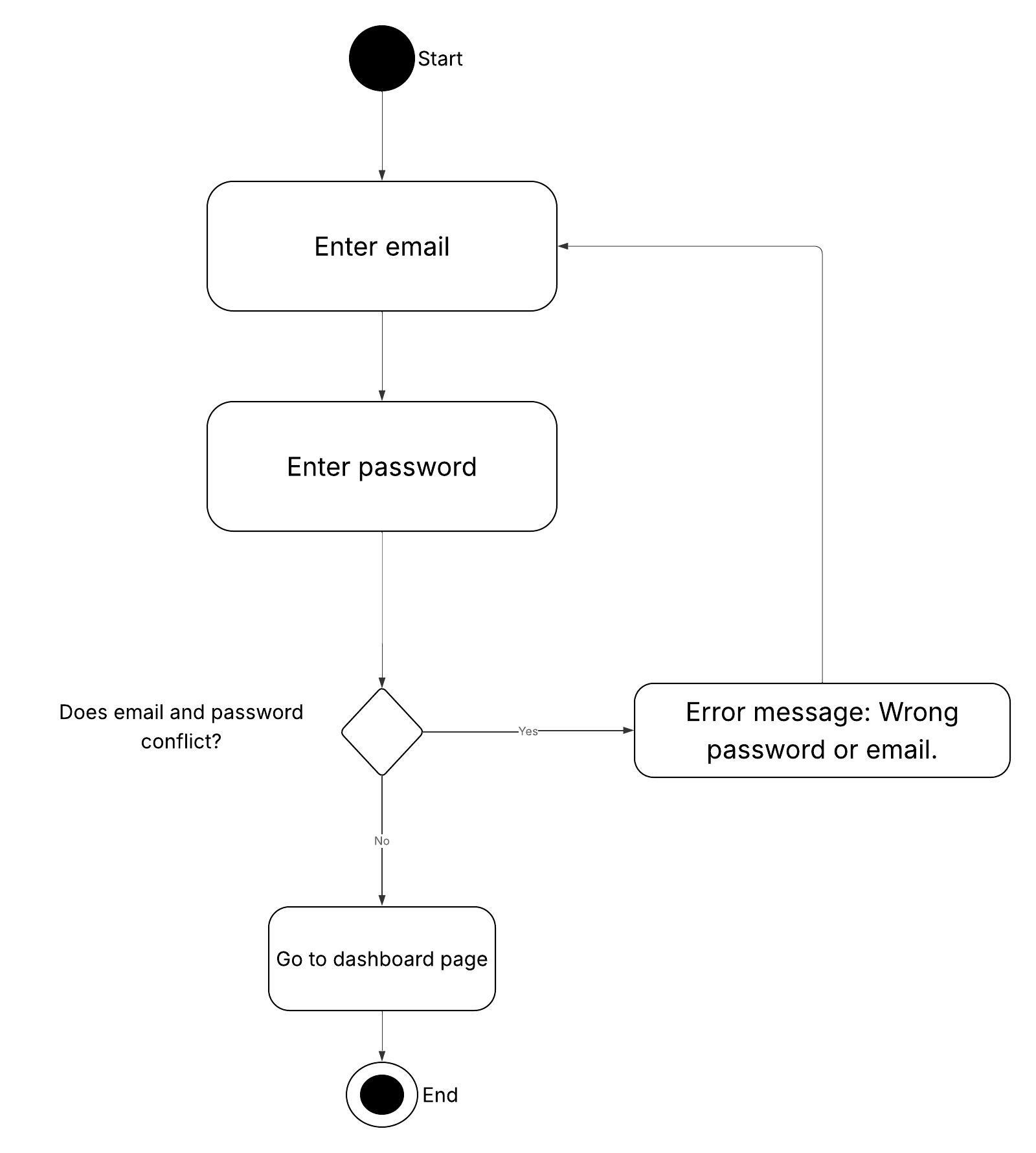
This feature enables users (admin or personnel) to access the system with their credentials. It is a **high priority** feature since it is the entry point of the system.

##### 3.2.1.2 Stimulus/Response Sequences

1. **User**: Enters username and password.
2. **System**: Validates credentials by checking the user database.  
     → If invalid, displays: “Incorrect username or password.”
3. **System**: If valid, redirects user to the appropriate home page based on their role (admin or personnel).

##### 3.2.1.3 Functional Requirements for Login

* **REQ-1**: The system shall provide input fields for username and password.
* **REQ-2**: The system shall verify the username and password against the user database.
* **REQ-3**: The system shall notify the user if the credentials are incorrect.
* **REQ-4**: The system shall redirect the user to the appropriate page upon successful login.
* **REQ-5**: The system shall support role-based access (admin/personnel).



### 3.2.2 Change Password Functionality

##### 3.2.2.1 Description and Priority

This feature allows users to change their current password after successful authentication. It is a **medium priority** feature for enhancing account security.

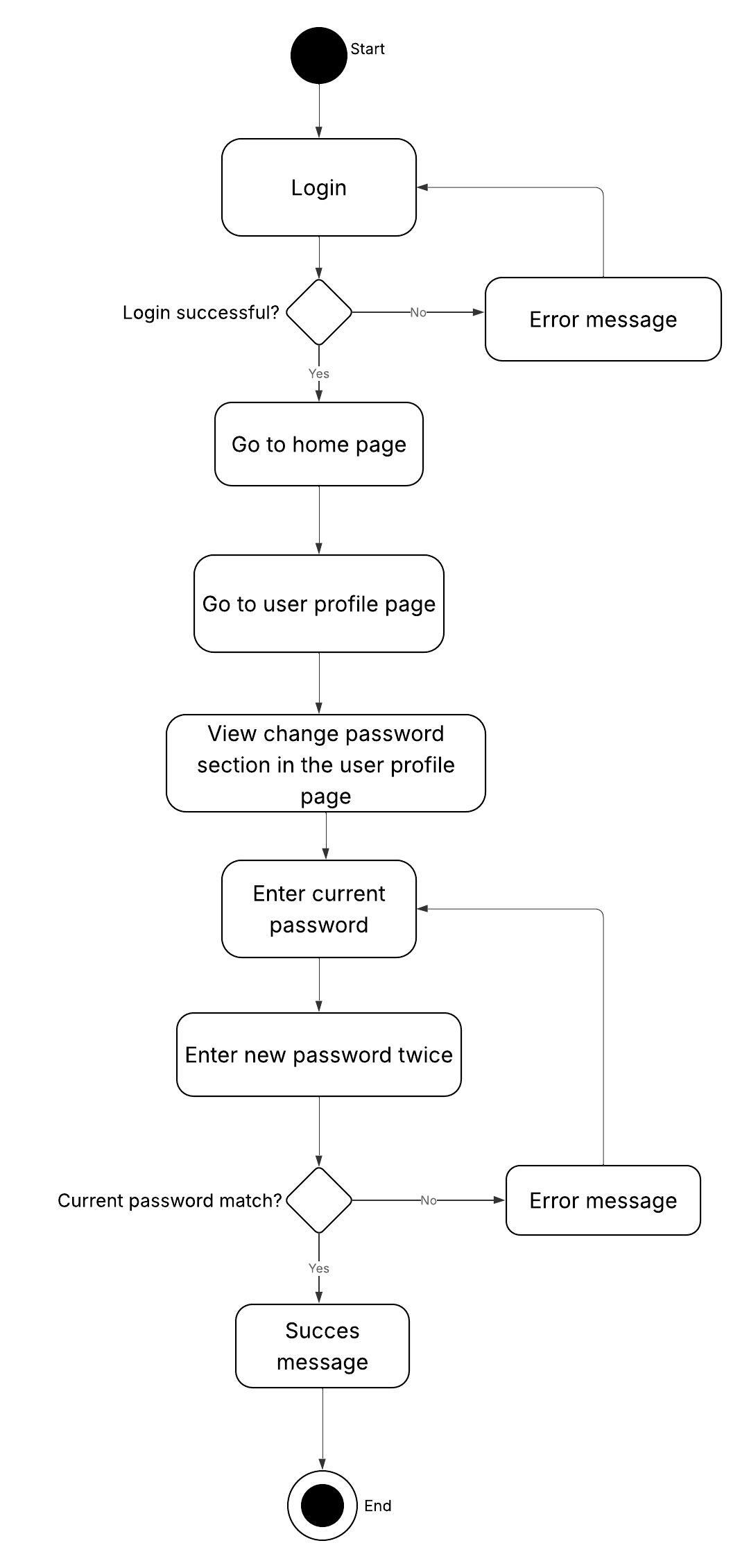
##### 3.2.2.2 Stimulus/Response Sequences

Sequence of user actions and system responses is listed below:

1. **User**: Navigate to “User Profile” page from the dashboard.
2. **User**: Enter current password.
3. **User**: Enter new password.
4. **User**: Re-enter new password for confirmation.
5. **System**: Validate that all fields are filled.
6. **System**: Check if the current password is correct.
7. **System**: Check if the new password and confirmation match.
8. **System**: If all validations pass, update the password and display success message.
9. **System**: If any validation fails, display appropriate warning or error message.

##### 3.2.2.3 Functional Requirements for Change Password

* **REQ-6**: User should be able to access the User Profile interface.
* **REQ-7**: System should require the user to input the current password.
* **REQ-8**: System should require the user to input and confirm a new password.
* **REQ-9**: System should verify the correctness of the current password.
* **REQ-10**: System should update the password if all validations pass.
* **REQ-11:** System should inform the user about success or failure of the operation.
* **REQ-12:** System should validate that all fields are filled.

****

### 3.2.3 Sample Adding and Listing

##### 3.2.3.1 Description and Priority

This feature allows the user to log into the system, navigate to the sample registration and listing page, add new samples, edit, delete, and filter them. Informational messages are displayed to improve user experience. This is a high-priority feature.

##### 3.2.3.2 Stimulus/Response Sequences

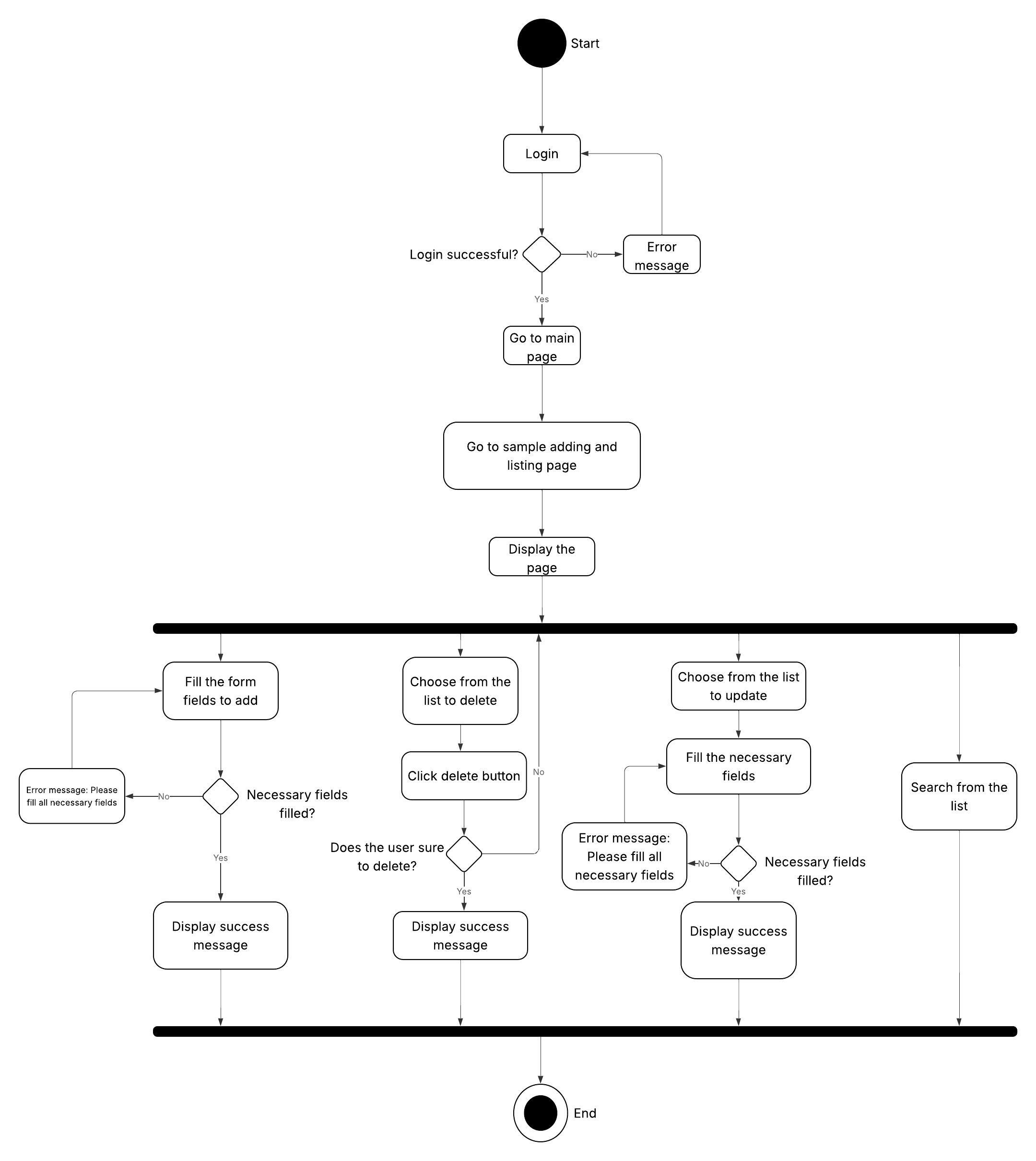
The sequence of user actions and system responses is as follows:

1. User: Logs into the system with valid credentials.
2. System: Validates login and opens the main page.
3. User: Clicks the "Sample Registration and Listing" button on the main page.
4. System: Opens the sample registration and listing page and displays the current samples in a list.
5. User: Fills in the form fields (business, project, sample name, cabinet, shelf number, entry date, weight, etc.) to add a new sample.
6. User: Clicks the "Save" button.
7. System: Validates the input, saves the new sample to the database.
8. System: Shows a confirmation message "Sample saved successfully."
9. System: Refreshes the sample list to include the new entry.
10. User: Selects a sample from the list and clicks the "Edit" button.
11. User: Modifies the sample details and clicks the "Update" button.
12. System: Updates the sample information in the database and shows a "Sample updated successfully" message.
13. System: Refreshes the list to reflect the updated data.
14. User: Selects a sample from the list and clicks the "Delete" button.
15. System: Displays a confirmation dialog "Are you sure you want to delete this sample?"
16. User:
    * If "Yes" is selected → System deletes the sample, shows "Sample deleted" message, and refreshes the list.
    * If "No" is selected → System closes the dialog and cancels the delete operation.
17. User: Enters filter criteria in the search box to filter the sample list.
18. System: Filters the sample list according to the criteria and updates the displayed list.

##### 3.2.3.3 Functional Requirements

The functional requirements related to this feature are listed below:

* **REQ-13**: User shall be able to log into the system.
* **REQ-14**: User shall be able to navigate to the "Sample Registration" page from the main page.
* **REQ-15**: System shall display the existing samples in a list.
* **REQ-16**: User shall be able to fill all necessary fields to add a new sample.
* **REQ-17**: System shall validate user input and display warning messages if required fields are missing or invalid.
* **REQ-18:** System shall display a confirmation message after a successful sample registration.
* **REQ-19:** User shall be able to edit existing sample details.
* **REQ-20:** System shall update sample data in the database and notify the user.
* **REQ-21:** User shall be able to delete a sample after confirmation.
* **REQ-22:** System shall delete the sample and display a confirmation message after user approval.
* **REQ-23:** User shall be able to filter the sample list based on criteria.
* **REQ-24:** System shall update the displayed sample list according to the filter.



### 3.2.4 Material Adding and Listing

##### 3.2.4.1 Description and Priority

This feature allows the user to add new materials, view and manage the list of existing materials, and perform update or delete operations. It is used to track material types and ensure data accuracy. This is a high-priority feature.

##### 3.2.4.2 Stimulus/Response Sequences

The interaction flow between the user and the system is as follows:

1. User: Logs into the system with valid credentials.
2. System: Authenticates the user and opens the main dashboard.
3. User: Clicks on the "Material Adding and Listing" button.
4. System: Opens the material adding and listing page and displays the existing material records in a table.
5. User: Enters material details (material name, cabinet, shelf number, entry date, number, description) into the registration form.
6. User: Clicks the "Save" button.
7. System: Validates the input fields.
8. System: If valid, saves the material to the database and shows a success message "Material saved successfully."
9. System: Refreshes the material list to include the new material.
10. User: Selects a material from the list and clicks the "Edit" button.
11. User: Modifies the material fields and clicks the "Update" button.
12. System: Validates the changes and updates the material in the database.
13. System: Displays a confirmation message "Material updated successfully" and refreshes the list.
14. User: Selects a material and clicks the "Delete" button.
15. System: Prompts with a confirmation dialog "Are you sure you want to delete this material?"
16. User:
    * If "Yes" → System deletes the material, shows "Material deleted" message, and updates the list.
    * If "No" → System cancels the delete operation.
17. User: Uses the search box to filter materials by name, code, or other fields.
18. System: Filters the material list accordingly and displays the results.

##### 3.2.4.3 Functional Requirements

The functional requirements related to this feature are listed below:

* **REQ-25**: User shall be able to log into the system.
* **REQ-26**: User shall be able to navigate to the "Material Adding and Listing" page from the main page.
* **REQ-27**: System shall display the existing materials in a list.
* **REQ-28**: User shall be able to fill all necessary fields to add a new material.
* **REQ-29**: System shall validate user input and display warning messages if required fields are missing or invalid.
* **REQ-30:** System shall display a confirmation message after a successful material registration.
* **REQ-31:** User shall be able to edit existing material details.
* **REQ-32:** System shall update material data in the database and notify the user.
* **REQ-33:** User shall be able to delete a material after confirmation.
* **REQ-34:** System shall delete the material and display a confirmation message after user approval.
* **REQ-35:** User shall be able to filter the material list based on criteria.
* **REQ-36:** System shall update the displayed material list according to the filter.

metin, diyagram, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

### 3.2.5 Stock Entry and Exit

##### 3.2.5.1 Description and Priority

This feature allows the user to manage the stock operations of samples by listing in/out records, searching existing in/out records, and adding new entries to them. It is a high-priority feature since it ensures accurate tracking of stock movement.

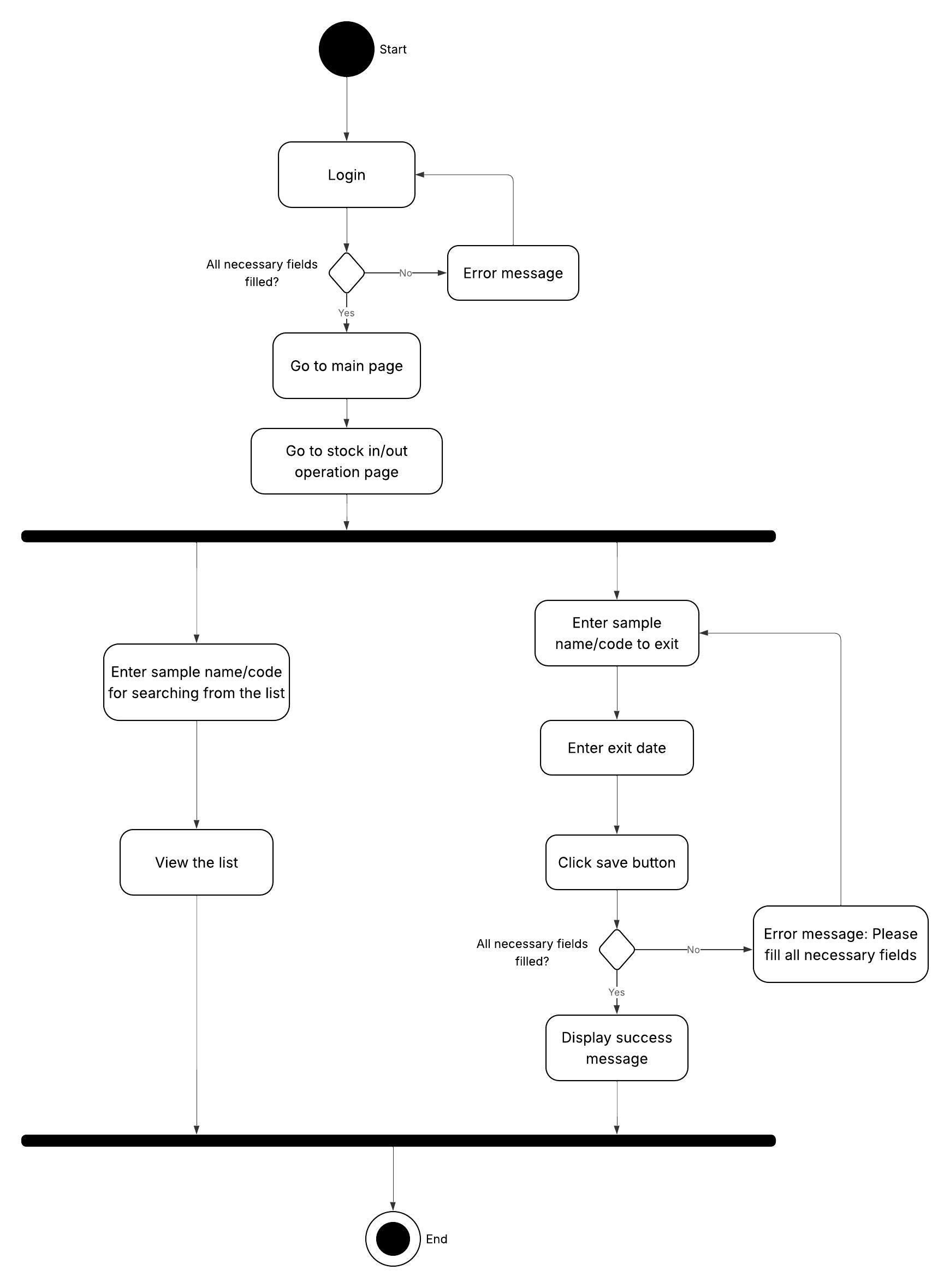
##### 3.2.5.2 Stimulus/Response Sequences

Sequences of user actions and system responses are listed below:

1. User: Logs into the system.
2. System: Verifies credentials and redirects the user to the homepage.
3. User: Clicks the "Stock In/Out Operations" button on the homepage.
4. System: Redirects to the Stock In/Out Operations page.
5. System: Displays a list of previous stock in/out records.
6. User: Searches a record by entering a sample name or other filtering criteria (optional).
7. System: Displays the filtered records based on search query.
8. User: Enters the sample name and selects the exit date in the provided form.
9. User: Clicks the "Save" button to record the operation.
10. System: Validates the input fields. If any required field is missing, displays a warning message (e.g., “Please enter sample name and select exit date.”)
11. System: If validation is successful, saves the new in/out record.
12. System: Updates the list with the new record.

##### 3.2.5.3 Functional Requirements for Stock In/Out Operations

* **REQ-37**: The system shall authenticate users before granting access to the homepage.
* **REQ-38**: The system shall redirect users to the homepage after a successful login.
* **REQ-39**: Users shall be able to access the Stock In/Out Operations page from the homepage.
* **REQ-40**: The system shall display a list of previously recorded stock in/out operations.
* **REQ-41**: Users shall be able to filter/search records by sample name or other criteria.
* **REQ-42:** The system shall show the filtered results dynamically based on user input.
* **REQ-43:** Users shall be able to add a new stock in/out entry by entering a sample name and selecting an exit date.
* **REQ-44:** The system shall validate required fields before saving a new record.
* **REQ-45:** If validation fails, the system shall display appropriate warning messages.
* **REQ-46:** The system shall save valid records and update the list dynamically.

****

### 3.2.6 Admin Operations

##### 3.2.6.1 Description and Priority

This feature allows the admin to manage user accounts by listing all users, editing or deleting existing users, and adding new users. It is a **high-priority** feature, as it enables effective control over system access and user roles.

##### 3.2.6.2 Stimulus/Response Sequences

Sequences of user actions and system responses are listed below:

1. **Admin:** Logs into the system.
2. **System:** Verifies credentials and redirects to the admin dashboard.
3. **System:** Displays a list of all registered users.
4. **Admin:** Clicks the "Edit" button next to a user.
5. **System:** Displays a form pre-filled with the user’s information.
6. **Admin:** Updates the user’s data and clicks the "Save" button.
7. **System:** Validates the input and updates the user’s information if valid.
8. **Admin:** Clicks the "Delete" button next to a user.
9. **System:** Displays a confirmation dialog.
10. **Admin:** Confirms the deletion.
11. **System:** Deletes the selected user and updates the list.
12. **Admin:** Fills out the form at the bottom of the page to add a new user.
13. **Admin:** Clicks the "Add User" button.
14. **System:** Validates the inputs. If required fields are missing, displays a warning message (e.g., “Please enter username and password.”)
15. **System:** If validation passes, adds the new user and refreshes the list.

##### 3.2.6.3 Functional Requirements for Admin Operations

* **REQ-47**: The system shall authenticate admins before granting access to the admin dashboard.
* **REQ-48**: The system shall allow access to the user management page via the dashboard.
* **REQ-49**: The system shall display a list of all registered users.
* **REQ-50**: The system shall allow the admin to edit user details.
* **REQ-51:** The system shall allow the admin to delete users after confirmation.
* **REQ-52:** The system shall update the user list after deletion.
* **REQ-53:** The system shall allow the admin to add a new user by filling in the required fields.
* **REQ-54:** The system shall validate all required fields before adding a new user.
* **REQ-55:** The system shall show appropriate warning messages if any validation fails.
* **REQ-56:** The system shall update the user list dynamically after adding a new user.
* **REQ-57:** The system shall validate edited user inputs before saving.

**metin, diyagram, taslak, çizim içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

# 4. Other Nonfunctional Requirements

## 4.1 Performance Requirements

1. The system shall respond to user actions (logging in, searching, saving data, monitoring) within 2 seconds under normal operating conditions.
2. The application shall launch and become ready for use within 5 seconds on recommended hardware.
3. Adding or deleting a record (e.g., a material, sample, or stock entry) shall be processed and reflected in the interface within 1 second.
4. The system shall support multiple user accounts, but only one user will operate per instance. No multi-user concurrency is required in this version.
5. The application shall perform efficiently with up to:
   1. 1,000 material records
   2. 500 sample entries
   3. 50 user accounts
6. The application shall use no more than:
   1. 500 MB RAM under normal operation
   2. 300 MB disk space for installation and database
7. On startup or when loading large lists (e.g., material inventory), the data shall be fully loaded and displayed within 3 seconds.

## 4.2 Security Requirements

1. The system must authenticate users through a secure login screen.
2. Each user must log in with a unique email.
3. Passwords must be stored using secure hashing algorithms.
4. Sensitive data should only be accessible to users with the appropriate permissions.
5. Unauthenticated users must not be able to access the application.
6. Only data modification is tracked.

## 4.3 Software Quality Attributes

1. The system must be user-friendly and intuitive, enabling users to navigate and complete tasks with minimal training or support.
2. The software should be easy to update, fix, and enhance. Code should be modular and well-documented to support future development and debugging efforts.
3. The system must deliver quick response times, maintain stable throughput, and handle expected load levels without performance degradation.
4. The system must be capable of handling increasing amounts of work or expanding to accommodate growth, both in terms of users and features.
5. The software should be able to operate with other systems, applications, or components as required, using standardized protocols and data formats.
6. The application should be easily deployable in different environments with minimal changes, including cloud and on-premises setups.
7. The system design should support easy testing, allowing for unit tests, integration tests, and system-level tests to be conducted efficiently.
8. The system should be accessible and operational during agreed-upon timeframes, ensuring minimal downtime and service interruptions.
9. Users should be guided through the system with informative warning messages, validation prompts, and error explanations to help them understand and correct their mistakes easily.