

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH  
UNIVERSITY OF INFORMATION TECHNOLOGY  
INFORMATION SYSTEM FACULTY**



**FINAL REPORT**

**SOFTWARE REQUIREMENT  
SPECIFICATION WITH DETON  
APPLICATION**

Lecturer: Dr. Nguyễn Tuấn Nam  
TA: Nguyễn Dương Tùng  
Class: CS4273.O12.CTTT  
Students: Phạm Đức Mạnh – 21522330  
Trần Hữu Mạnh – 21522331  
Lý Tấn Phát – 21522444  
Đặng Hữu Quang – 21552508  
Nguyễn Đình Viễn Sơn – 21522555  
Nguyễn Minh Thông – 21522644

# Software Requirements Specification

---

## Deton

Group:

Phạm Đức Mạnh – 21522330

Trần Hữu Mạnh – 21522331

Lý Tấn Phát – 21522444

Đặng Hữu Quang – 21552508

Nguyễn Đình Viễn Sơn – 21522555

Nguyễn Minh Thông – 21522644

# Table of Contents

1. Introduction .....	5
1.1 Purpose .....	5
1.2 Scope .....	5
1.3 Definitions, acronyms, and abbreviations .....	6
1.4 Reference .....	7
1.5 Overview .....	7
2. Overall description .....	7
2.1 Product perspective .....	7
2.2 Product function .....	8
2.3 User characteristics.....	8
2.4 Constraints.....	8
2.5 Assumptions and Dependencies .....	9
3. Specific requirements .....	9
3.1 External interface requirements.....	9
3.1.2 Hardware interfaces.....	9
3.1.3 Software interfaces .....	9
3.1.1 User interfaces .....	10
3.2 Functional requirements .....	11
3.2.1 Sign up.....	11
3.2.2 Log in.....	14
3.2.3 Create a new project. ....	18
3.2.4 Open a project.....	21
3.2.5 Delete a project.....	24
3.2.6 Update project. ....	26
3.3 Nonfunctional requirements .....	30

3.3.1 Usability .....	30
3.3.2 Performance .....	30
3.3.3 Security .....	30
3.3.4 Audit .....	31
3.3.5 Maintainability .....	31
4. Other requirements .....	32
4.1 Database .....	32
4.2 MyPaint .....	32
Appendix A: Glossary .....	32
Appendix B: Analysis Models.....	33

## **1. Introduction**

Deton is a cutting-edge management drawing application designed to streamline the creation, editing, and organization of drawings across diverse disciplines. Its robust organizational tools enable efficient categorization and storage of drawings, promoting enhanced productivity and workflow efficiency. Ensuring compatibility with different file formats and integrating smoothly with existing tools, Deton prioritizes data security through encryption methods and access controls, reinforcing its reliability in safeguarding sensitive intellectual property.

The SRS document further includes a section dedicated to defining abbreviations and specialized terms used throughout the document, aiding in comprehensive understanding of Deton's specifications and functionalities.

### **1.1 Purpose**

The purpose of Deton is to provide a sophisticated and user-friendly platform that simplifies the intricate process of creating, managing, and collaborating on drawings across diverse professional domains.

By offering this document is to give a detailed description of the requirements for the “Deton” software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface, and functions. This document is primarily intended to show stakeholders how the application works and meets its requirements.

### **1.2 Scope**

The “Deton” is an application, which has the normal function of a painting application such as using a brush, changing color, rotating,..., created to help users to make the “paint” become more convenient and flexible with the ability to save and store their work on the database. Users will be able to organize their work better with the accounting system.

Furthermore, the application does not need the Internet to be able to log in to your account because the database is created locally which is on your computer. The application should be free to download and accessible to any user who uses PC-based devices.

### 1.3 Definitions, acronyms, and abbreviations

<b>Term</b>	<b>Definition</b>
User	Someone who interacts with the application
Stakeholder	Any person who interacts with the system who is not a developer
User ID	A unique identifier is assigned to each user
DESC	Description
RAT	Rational
DEP	Dependency
BR	Business rule
TAG	A unique, persistent identifier contained in a PLanguage statement [1]
GIST	A short, simple description of the concept contained in a PLanguage statement [1]
SCALE	The scale of measure used by the requirement contained in a PLanguage statement [1]
METER	The process or device used to establish a location on a SCALE contained in a PLanguage statement [1]
MUST	The minimum level required to avoid failure contained in a PLanguage statement [1]

## 1.4 Reference

[1] Feldt R, "re\_lecture5b\_100914", unpublished.

[2] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[3] Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, January 2005.

## 1.5 Overview

The remainder of this document includes four chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. Further, the chapter also mentions the system constraints and assumptions about the product. The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used to specify the requirements more precisely for different audiences. The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen. The appendixes at the end of the document include all the results of the requirement prioritization and a release plan based on them.

## 2. Overall description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce its basic functionality. At last, the constraints and assumptions for the system will be presented.

### 2.1 Product perspective

This system will consist of two parts: the **application** and the **database**.

The application not only has the normal paint function with various tools such as eraser, pencil, or selection tool but also the log-in and sign-up functionality which can be used for managing the work of the user. The user will be able to modify multiple works distinguished from others by name.

To do that, the application will need to connect to the local database, which contains the account information such as user ID, username, and password. This information will be used to demonstrate each particular user. When the user signs up for the new account, the database will be modified and the account can be used to log in for the next time. The work can also be saved as an image and stored on the user's devices.

## **2.2 Product function**

The application will provide drawing and painting tools including brush, pencil, and other drawing tools, creating shapes, lines, and text. Users can also change the tool output color by color palette, which offers a color palette with a wide range of colors and gradients, allowing users to customize and save their favorite colors. Not only that, selection and transformation tools are also available for selecting and transforming portions of the artwork which include options for resizing, rotating, and flipping selected elements.

They are exporting artworks in various file formats (JPEG, PNG, and JPG). It also has an intuitive and user-friendly interface for easy navigation ensuring accessibility for both novice and experienced users. Provide user preferences for customizing the interface, shortcuts, and other settings and allow users to save and load their custom workspace configurations. Include help resources, instructions, and documentation to assist users in learning and mastering the app's features.

## **2.3 User characteristics**

There is only one type of user that interacts with the system that is who directly uses the application. They can only use the function which I have already mentioned before. Because the management of the users is not necessary, the administrators do not need to be included in the application. However, if the application continues to be developed the online functionality more stakeholders might appear in the future.

## **2.4 Constraints**

One of its biggest constraints is the platform it can run on, which only includes PC-based devices. Another problem is that it is developed from C# Winform which is to say that it is a very old tool to develop an application so the application's performance has been affected heavily.

Not only that, but security can also be a problem because we don't have administrators to manage user-created content, and also the reset password function.



Because of that, if the user has a problem with their account or they don't remember their password, the work they made might be lost forever.

## **2.5 Assumptions and Dependencies**

One assumption about the product is that users have a basic understanding of digital painting concepts and tools. The components also work in the same way, if a different user's device has a different interface then the application has to be specifically adjusted to each interface which can lead to different requirements.

The application also depends on the accuracy of the documentation which helps users to understand and use the product better.

## **3. Specific requirements**

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

### **3.1 External interface requirements**

This section provides a detailed description of all inputs into and outputs from the system. It also describes the hardware, software, and communication interfaces and provides basic prototypes of the user interface.

#### **3.1.2 Hardware interfaces**

Since the application doesn't have any designated hardware, it doesn't have any direct hardware interfaces. The network is also the same while the app doesn't have the requirement for communication over a network.

#### **3.1.3 Software interfaces**

The application connects with the database to get the authorization for logging in or validation for signing up. It also stores the project of that's person account, so it consists of operations concerning both reading and modifying the data.

### 3.1.1 User interfaces

At first, when the user runs the application, the log-in page will appear as in Figure 1. If they already have an account, they can fill it in the form. On the other hand, if the user does not have an account, they can click on the “Create account” button then the sign-up page will appear as in Figure 2. The form will appear and users can fill it out to create a new account.

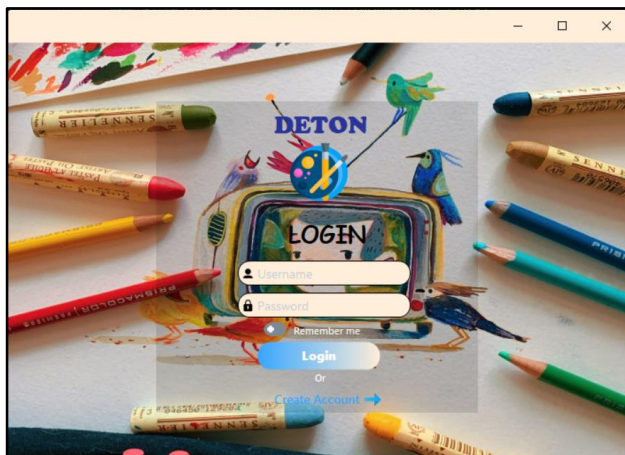


Figure 1: Log-in page

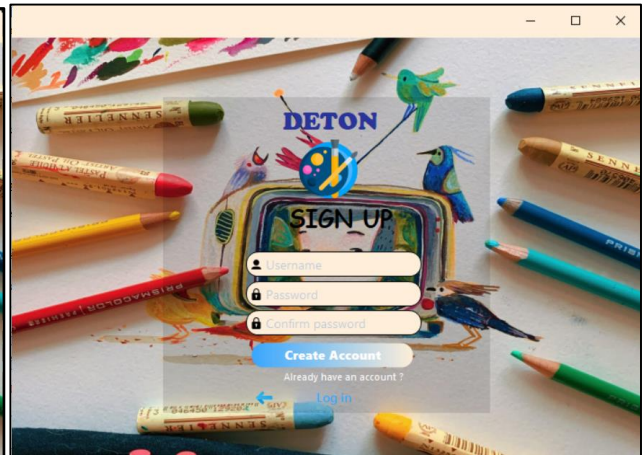


Figure 2: Sign-up page

After successfully logging into the account, the user will be able to see the Homepage as in Figure 3, this is where they interact with their works. Users can choose existing projects or create a new one by clicking the panel has the “+” sign in the middle. After doing that, the painting page will appear as in Figure 4 where users use to make changes to their painting by choosing various tools in the column on the left of the screen.

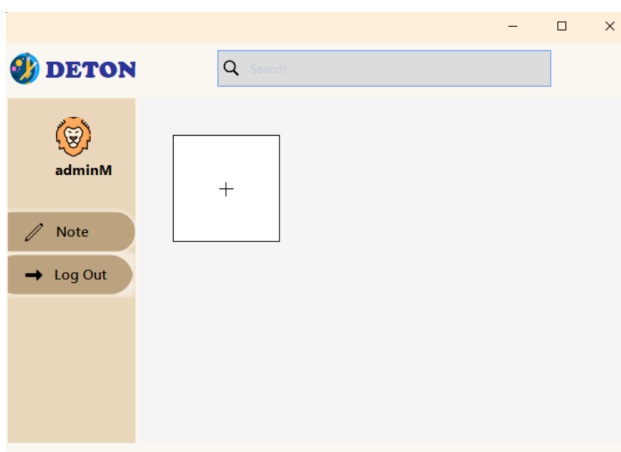


Figure 3: Painting page

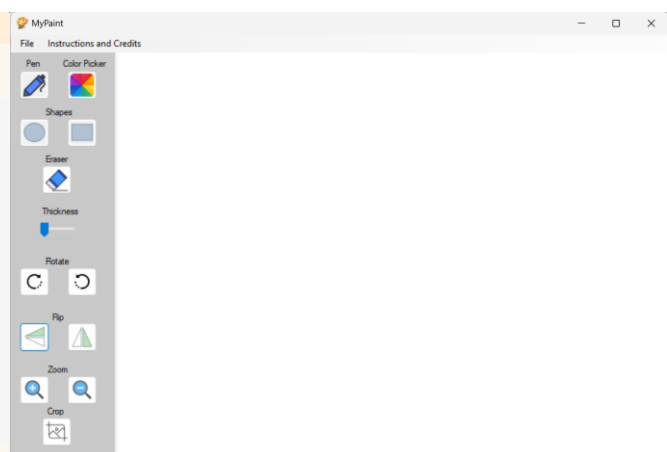
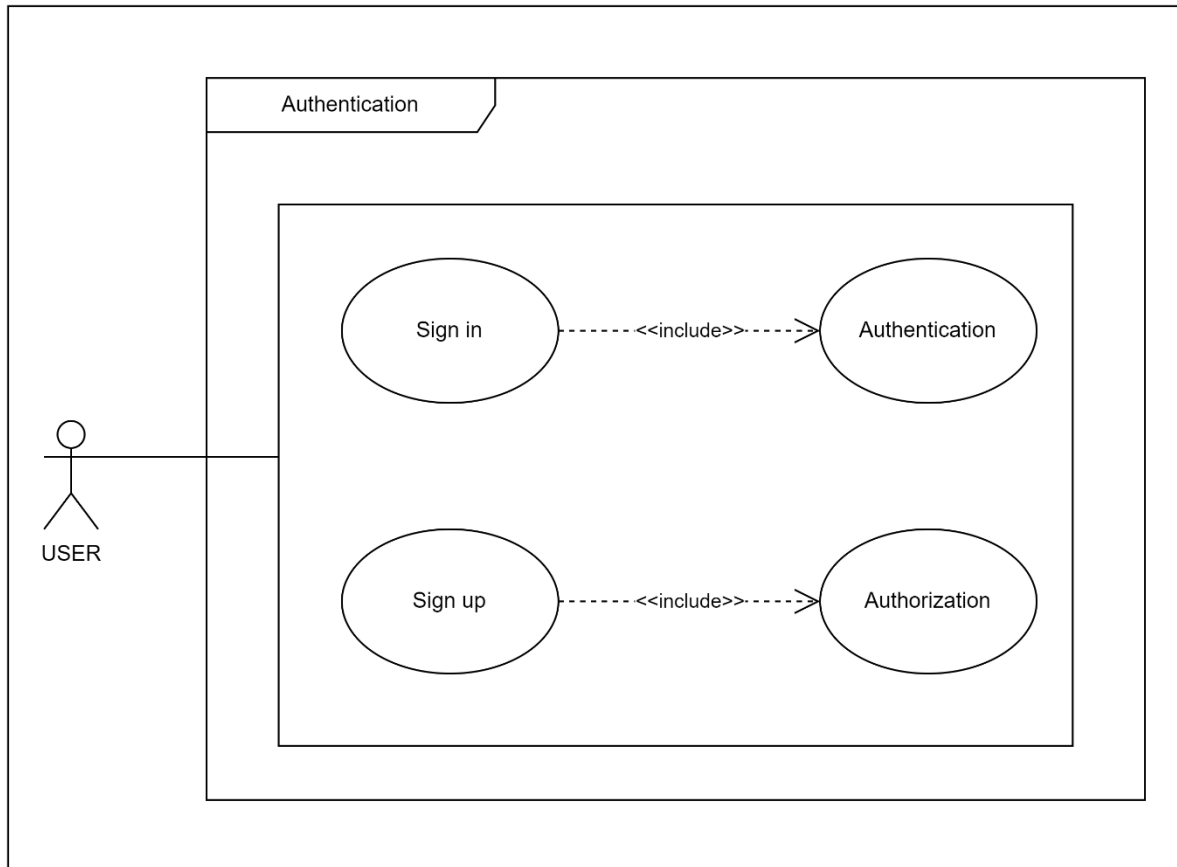


Figure 4: Homepage

## 3.2 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software system.



### 3.2.1 Sign up

**ID: FR\_1**

**TITLE:** User Registration.

**DESC:** A user should have the ability to sign up for an account through the application.

**BR:**

- Username field: The user must enter a unique username and must not leave it blank.
- Password field: The user must not leave it blank.
- Confirm Password field: The user must enter the same info as in the password field.

**RAT:** For users to have an account and use the app.

**DEP:** A database for processing and storing user registration information.

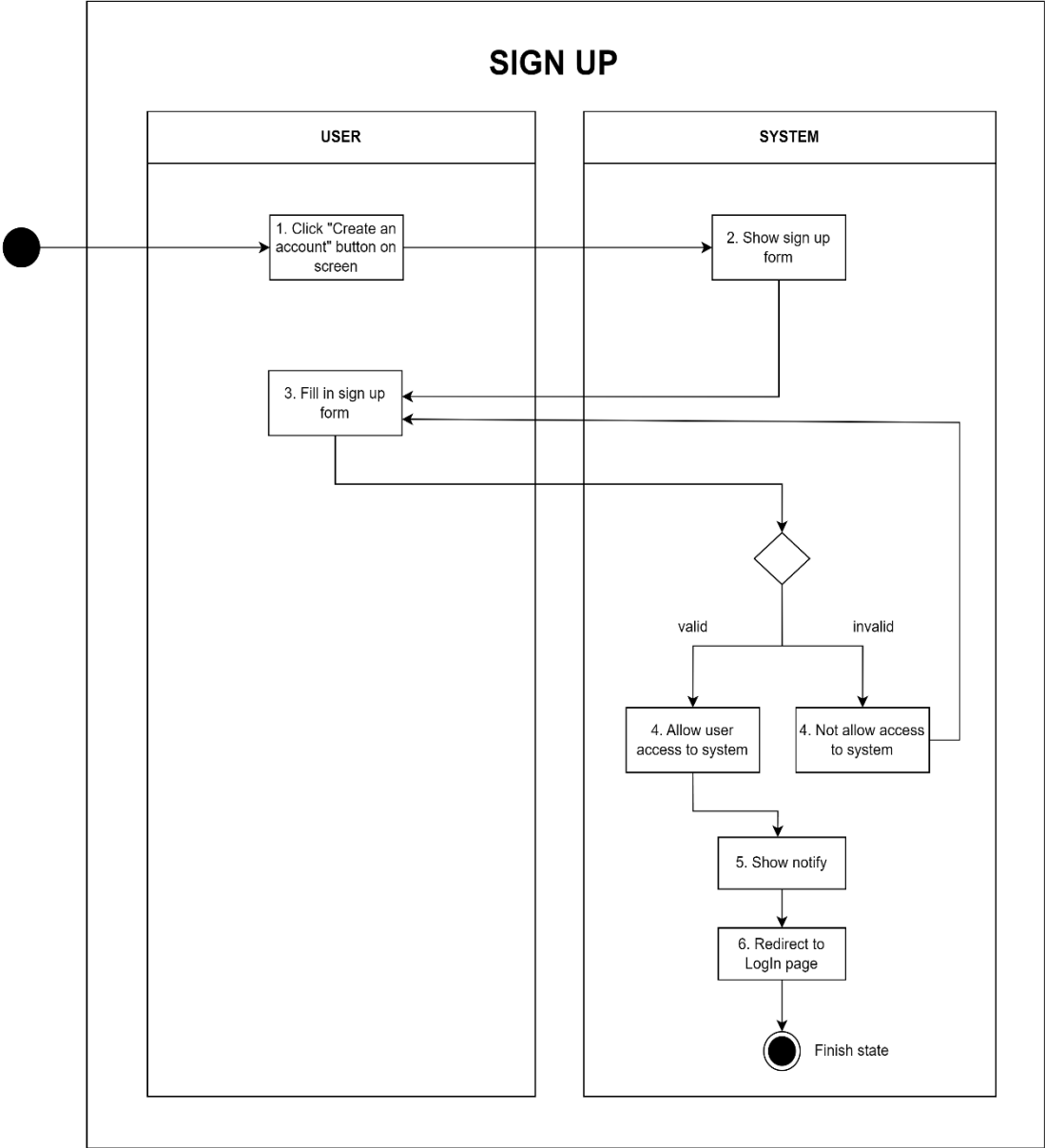
This is the specification of the sign-up use case:

Specification: ID: **FR\_1**

<b>Name</b>	<b>Sign-up (ID: FR_1)</b>
Scenario	Users want to create account to use the services
Description	This use-case occurs when users desire to create an account to access the system. After signing up successfully, users are directed to the home page
Actor	User
Trigger	The user clicks on the sign-up button
Related use-case	Authorization
Pre-conditions	The user opens the application and clicks on the “Create Account” button
Post-conditions	Success: The user has created an account successfully. Fail: The user fails to create an account
Basic events	<ol style="list-style-type: none"><li>1. Users open the application</li><li>2. Users click on the sign-up button</li><li>3. Users fill in username and password</li><li>4. System check the information</li><li>5. The system creates the account</li></ol>
Alternative events	4a. If the Information is valid, continue to step 5.
Exception conditions	4b. If the information is already in the system, the system responds to the error message.

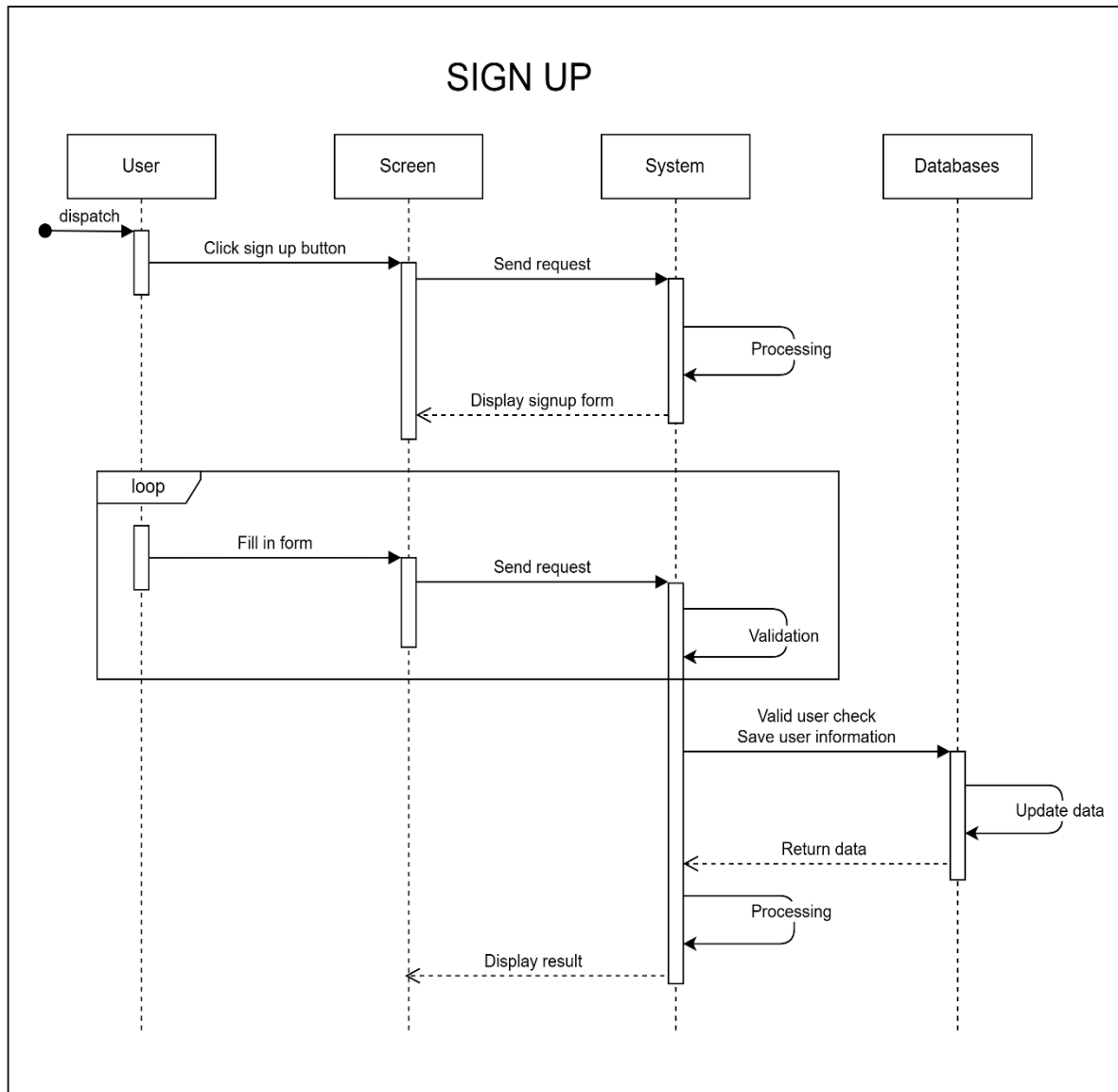
This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:

### Activity diagram



This sequence diagram will represent the interactions between different components or objects within a system and the database:

### Sequence diagram



### 3.2.2 Log in

**ID: FR\_2**

TITLE: User Login.

DESC: Given that a user has registered, then the user should be able to log in to the application. The log-in information will be stored in the database.

BR:

- Username field: The user must not leave it blank.

- Password field: The user must not leave it blank.

RAT: For a user to log in to the application.

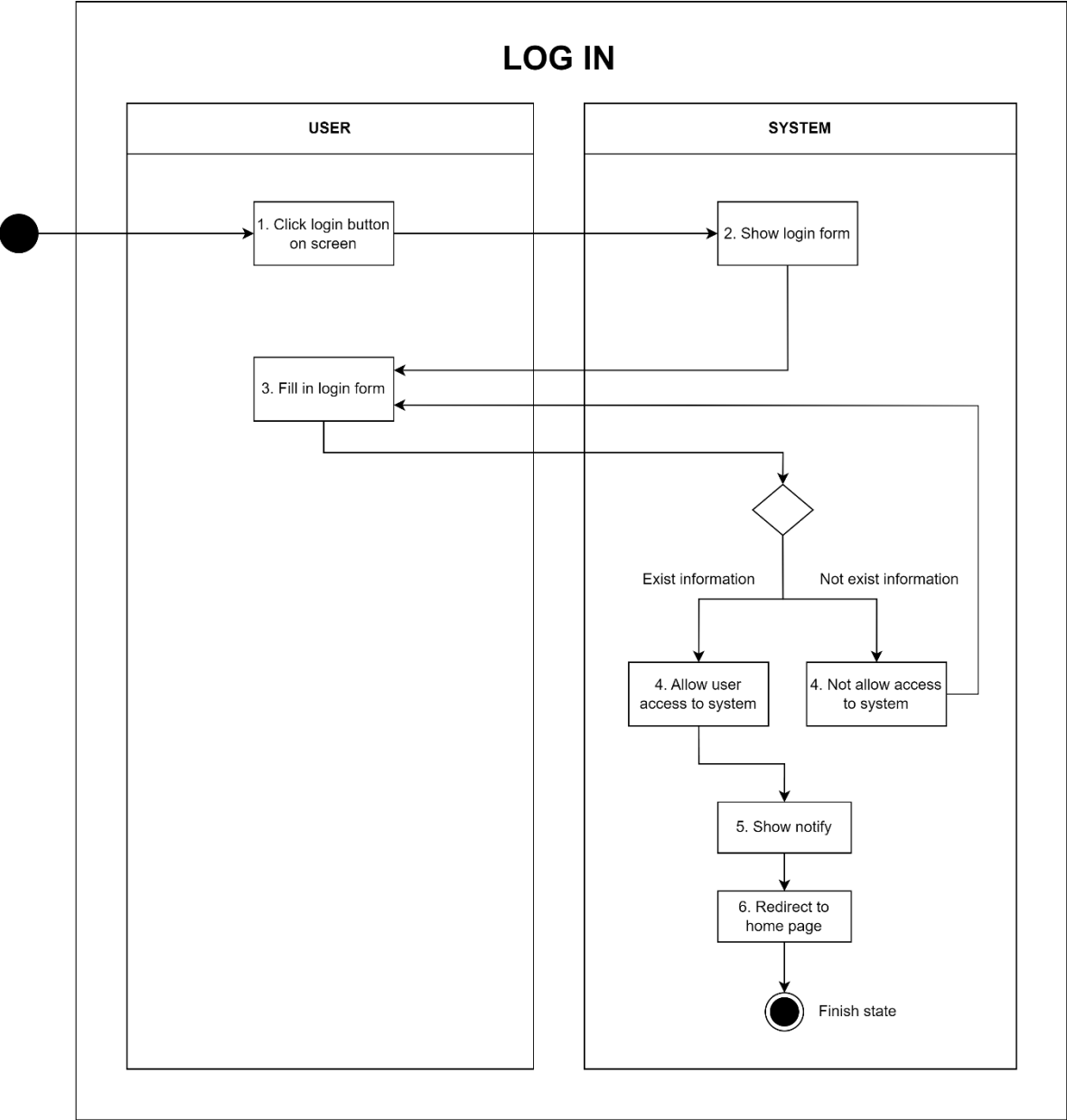
DEP: FR\_1.

This is the specification of the sign-in use case:

Name	Log in <i>FR_2</i>
Scenario	Users want to access the system to use the services.
Description	This use-case occurs when users desire to access the system. After logging successfully, users are directed to the default page.
Actor	User
Trigger	Users fill in the username and password fields. Then press the login button.
Related use-case	Authentication
Pre-conditions	The user opens the application
Post-conditions	Success: Users have system access rights. Fail: Deny access request, showing error messages before starting to sign in again
Basic events	<ol style="list-style-type: none"> <li>1. Users open applications.</li> <li>2. Users input username and password.</li> <li>3. Users press the “Log in” button.</li> <li>4. The system checks the validity of the user’s information</li> <li>5. System responds to the successful message and redirects to default page systems.</li> </ol>
Alternative events	4.a. If information is valid, continue to step 5
Exception conditions	4.b.1. If information is invalid, the system responds to the error message, rejects the request, and states on the login page.

This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:

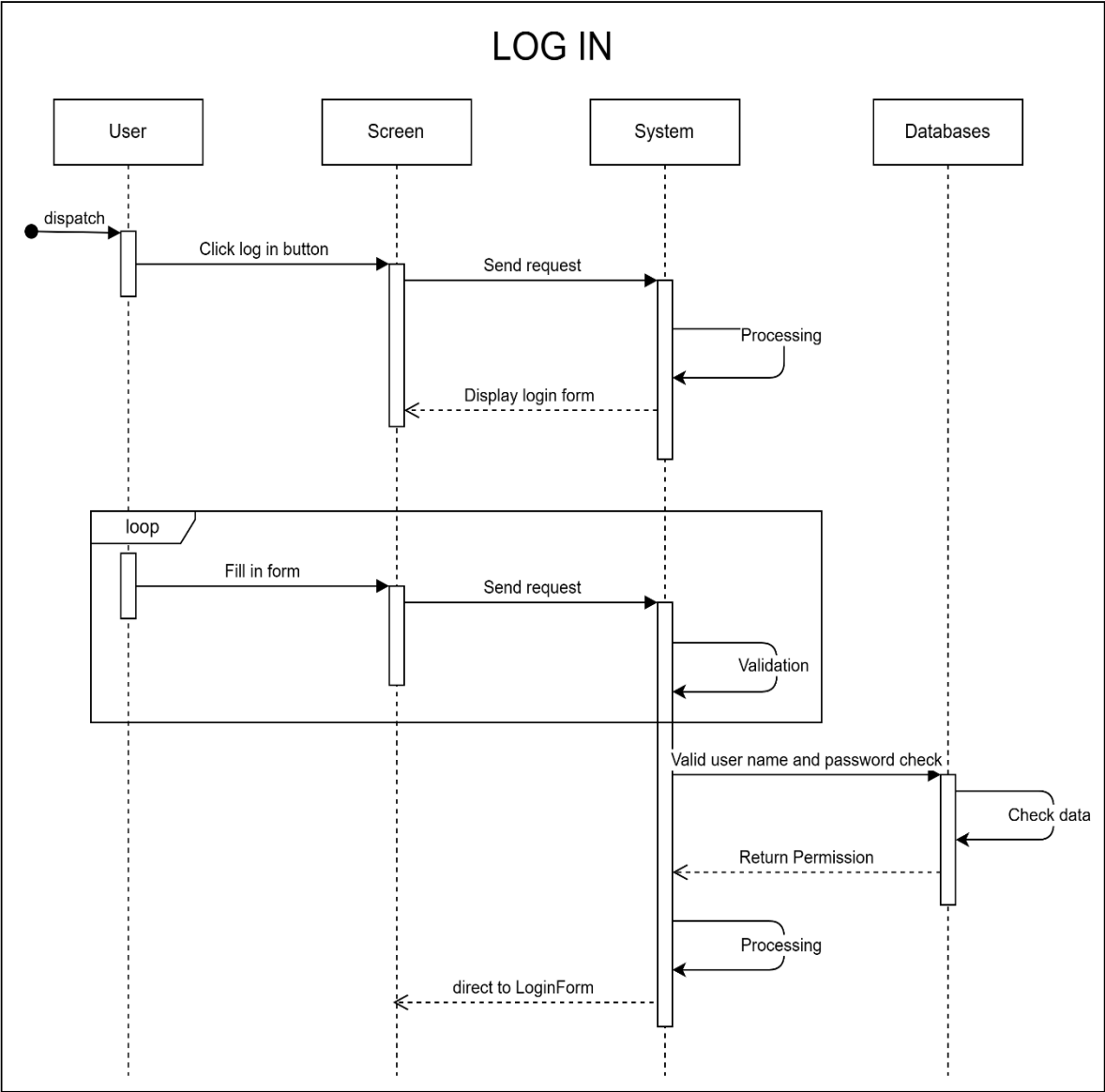
### Activity diagram

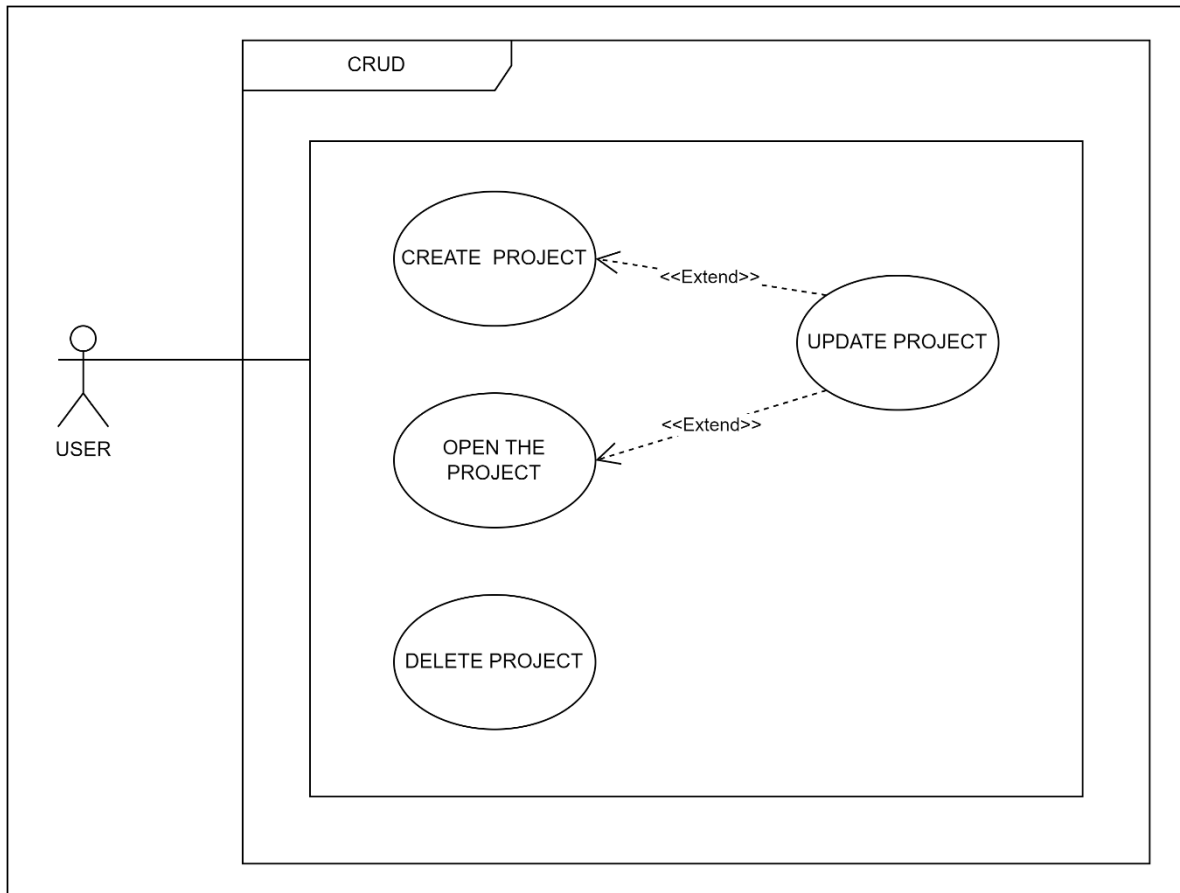




This sequence diagram will represent the interactions between different components or objects within a system and the database:

### Sequence diagram





### 3.2.3 Create a new project.

**ID: FR\_3**

**TITLE:** Create a new project.

**DESC:** Given that a user has logged in to the application, then the user should be able to create a project to draw.

**BR:**

- The user clicks on the “+” on the home screen.
- A form appears, and the user has to name the file and then click the “OK” button.

**RAT:** For a user to take notes.

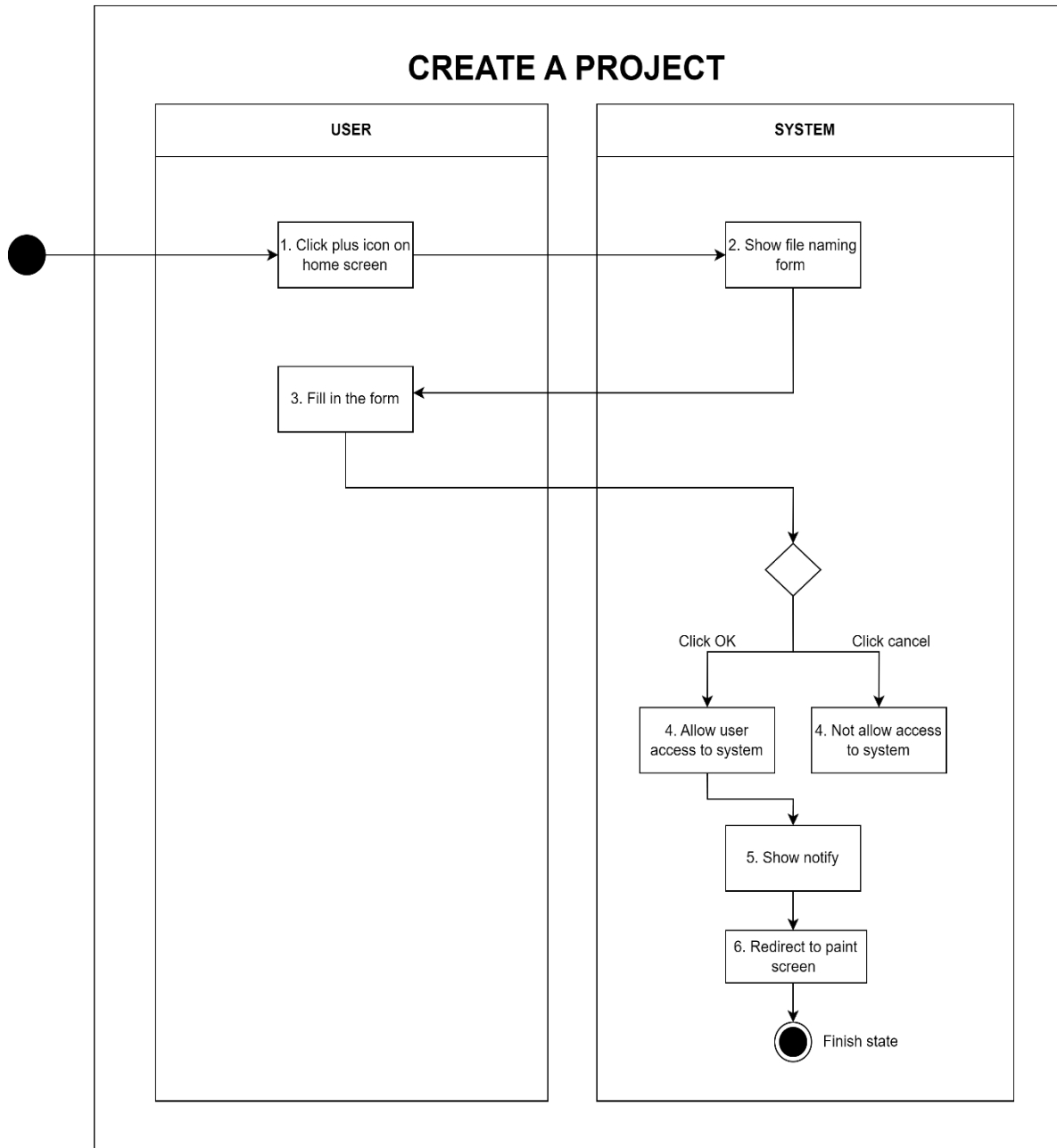
**DEP:** FR\_2.

This is the specification of the Create project use case:

Name	Create project (FR_03)
Scenario	Creating a New Project in the application.
Description	This occurs when the user wants to create a new project. After the project is created successfully, the user can use the application function.
Actor	User
Trigger	The user logged in to the application and clicked the “+” button.
Related use-case	Update project.
Pre-conditions	Users logged in to the application successfully
Post-conditions	Success: A new project created Fail: Showing error message
Basic events	<ol style="list-style-type: none"> <li>1. User logged in to the application</li> <li>2. The user clicked the “New project” button to create a new project</li> <li>3. User choose the name of the project</li> <li>4. The system created that project</li> </ol>
Alternative events	3a. If the request is valid, continue step 4
Exception conditions	<p>3b1. The application fails to respond or crashes during the project creation process.</p> <p>3b2. The user did not fill in anything</p>

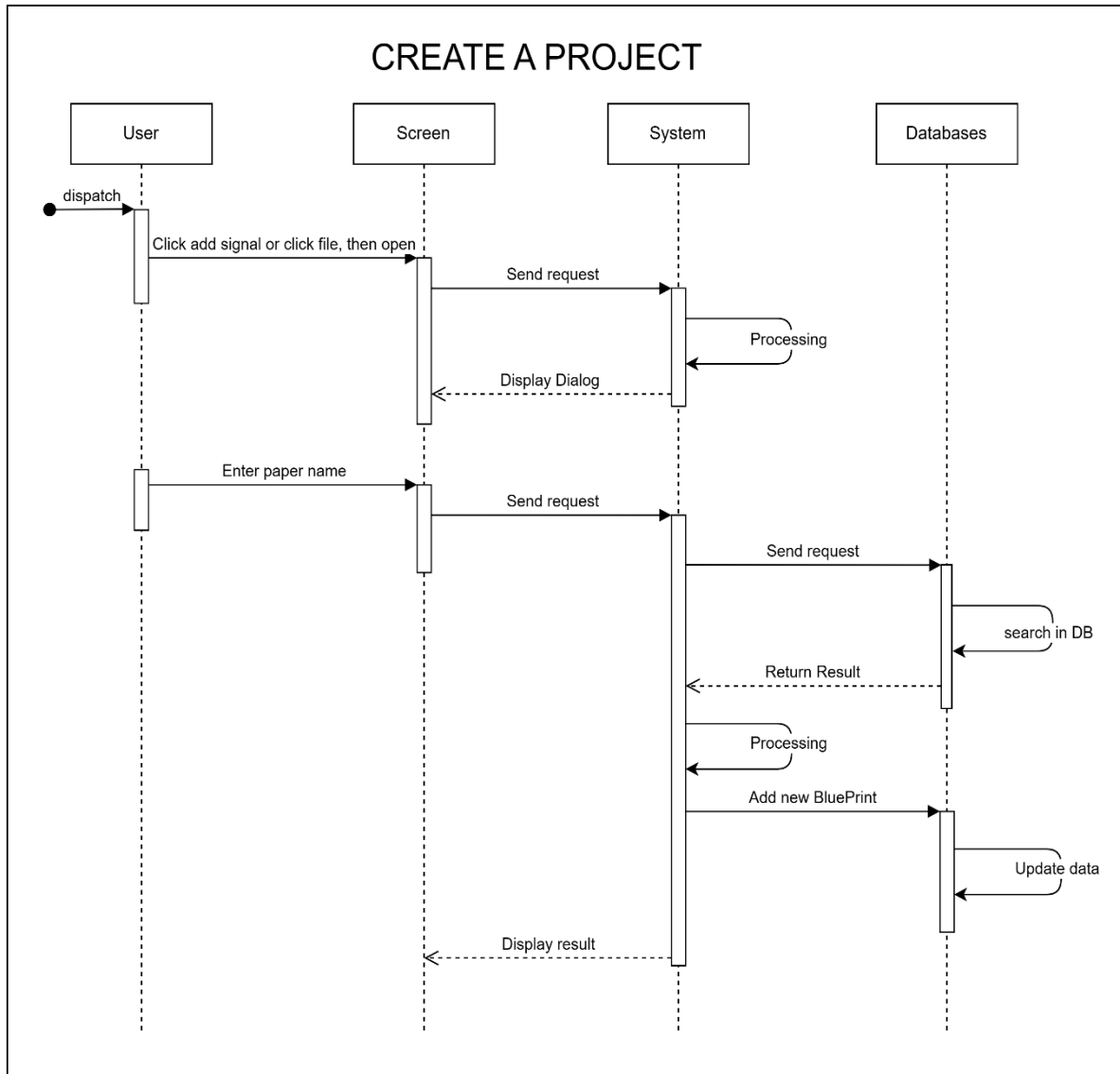
This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:

### Activity diagram



This sequence diagram will represent the interactions between different components or objects within a system and the database:

### Sequence diagram



#### 3.2.4 Open a project.

ID: **FR\_4**

TITLE: Open a project.

DESC: Given that a user has logged in to the application, then the user must have the ability to open an existing project in their account and devices.

BR:

- The user clicks “Open” in the file menu or presses the key Ctrl + O.
- User selects the file they want to open in their account.

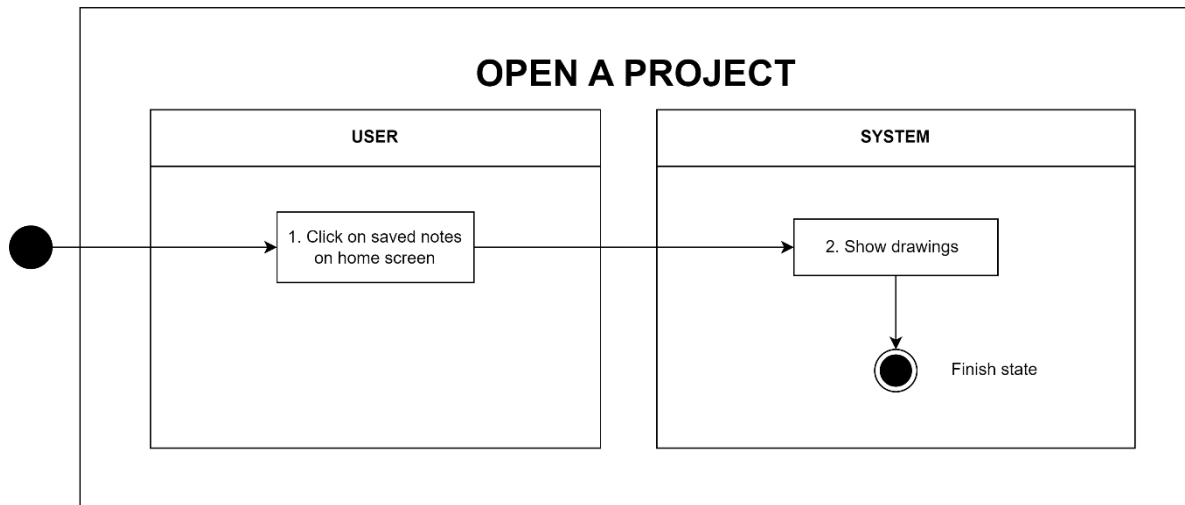
RAT: For a user to open files.

DEP: FR\_3.

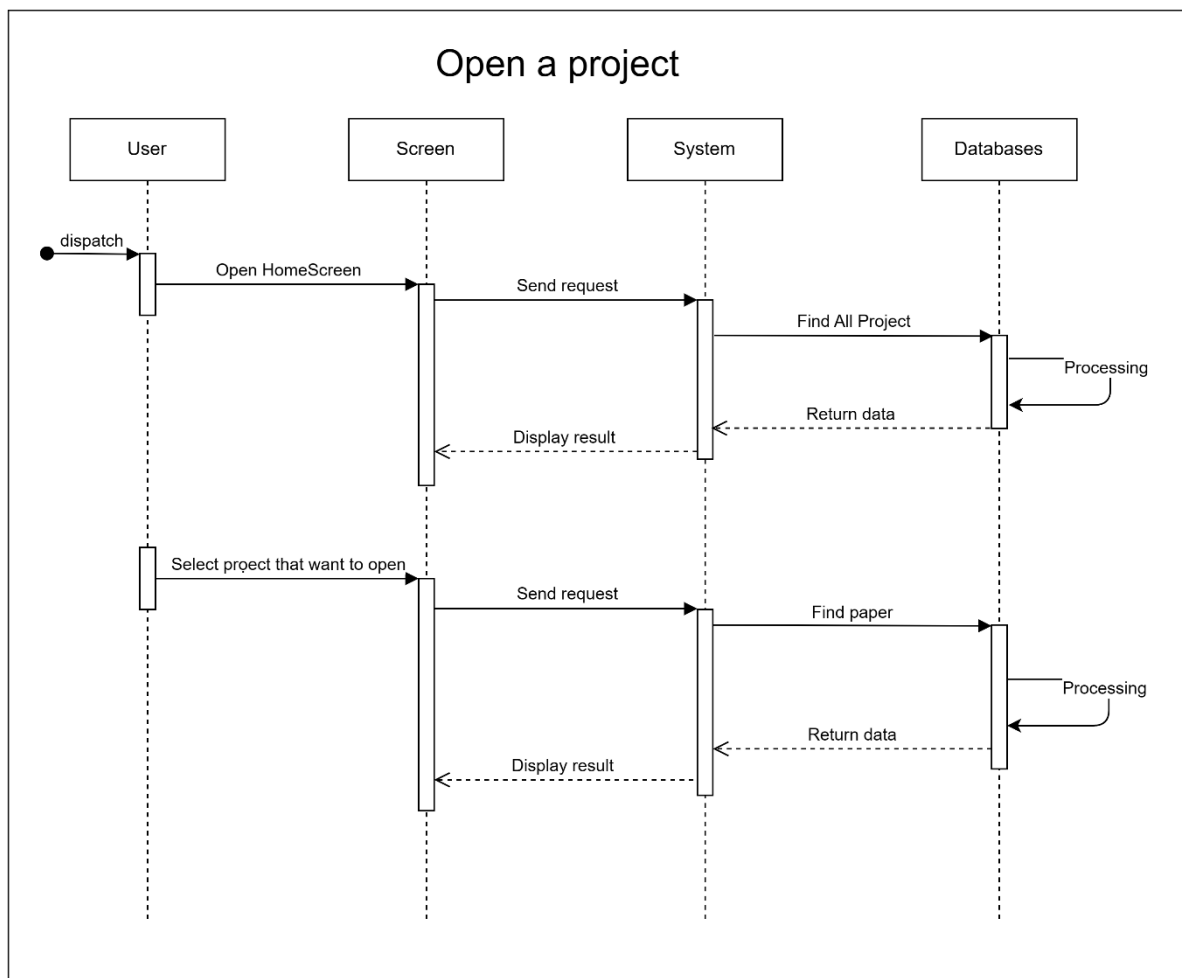
This is the specification of the Open project use case:

Name	Open project (FR_4)
Scenario	Opening an Existing Project on the computer.
Description	This use case occurs when the user wants to open a project that is already on the computer.
Actor	User
Trigger	The user logs in and clicks on the existing file.
Related use-case	Update project.
Pre-conditions	Users must be logged into the system successfully.
Post-conditions	Success: Users open the file successfully Fail: Deny opening the file
Basic events	1. User logged in to the application 2. User click on the file that already had 3. The system shows the file
Alternative events	2a. If the request is valid, continue step 3. 3a. The user can still open other files
Exception conditions	2b1. File not found. 2b2. Unsupported file format 2b3. File corruption

This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:



This sequence diagram will represent the interactions between different components or objects within a system and the database:



### 3.2.5 Delete a project.

**ID: FR\_5**

TITLE: Delete paper.

DESC: Given that a user has logged in to the application, then the user must have the ability to delete an existing paper in their account.

BR:

- The user clicks the “X button” in the corner of the file.
- User confirms whether they want to delete or not.

RAT: For a user to delete files.

DEP: FR\_2.

This is the specification of the Delete project use case:

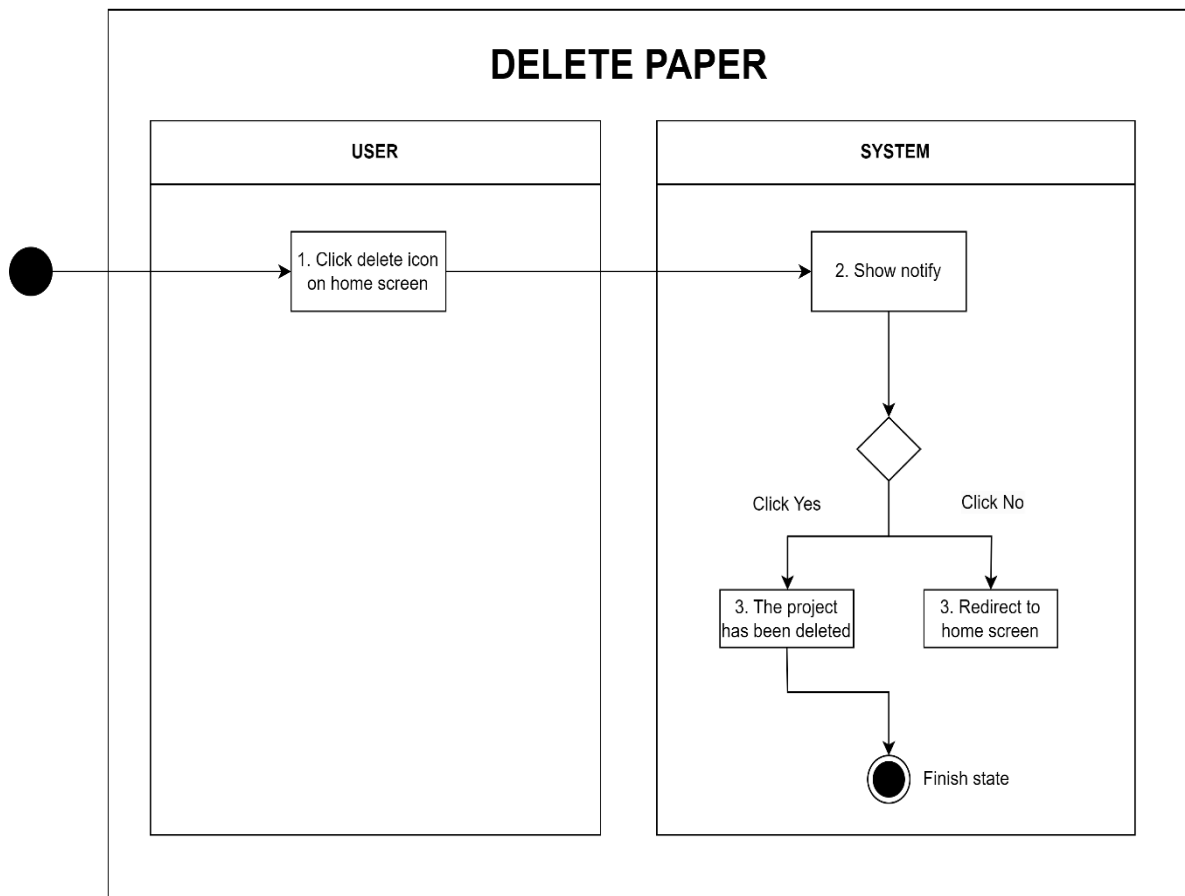
Name	Delete project (FR_5)
Scenario	The user wants to delete the project.
Description	This use case occurs when the user wants to delete the project that the user does not need anymore
Actor	User
Trigger	The user logged in to the system and click to delete the project
Related use-case	
Pre-conditions	Users must be logged into the system successfully.
Post-conditions	Success: The selected project is permanently removed from the file system. Fail: Deny request, showing error message
Basic events	1. Logging in to the application 2. Choosing the file to delete then click on the “trash can” button to permanently delete the file 3. The file was deleted successfully



Alternative events	2a. Request is valid, continue step 3
Exception conditions	2b1. File in use

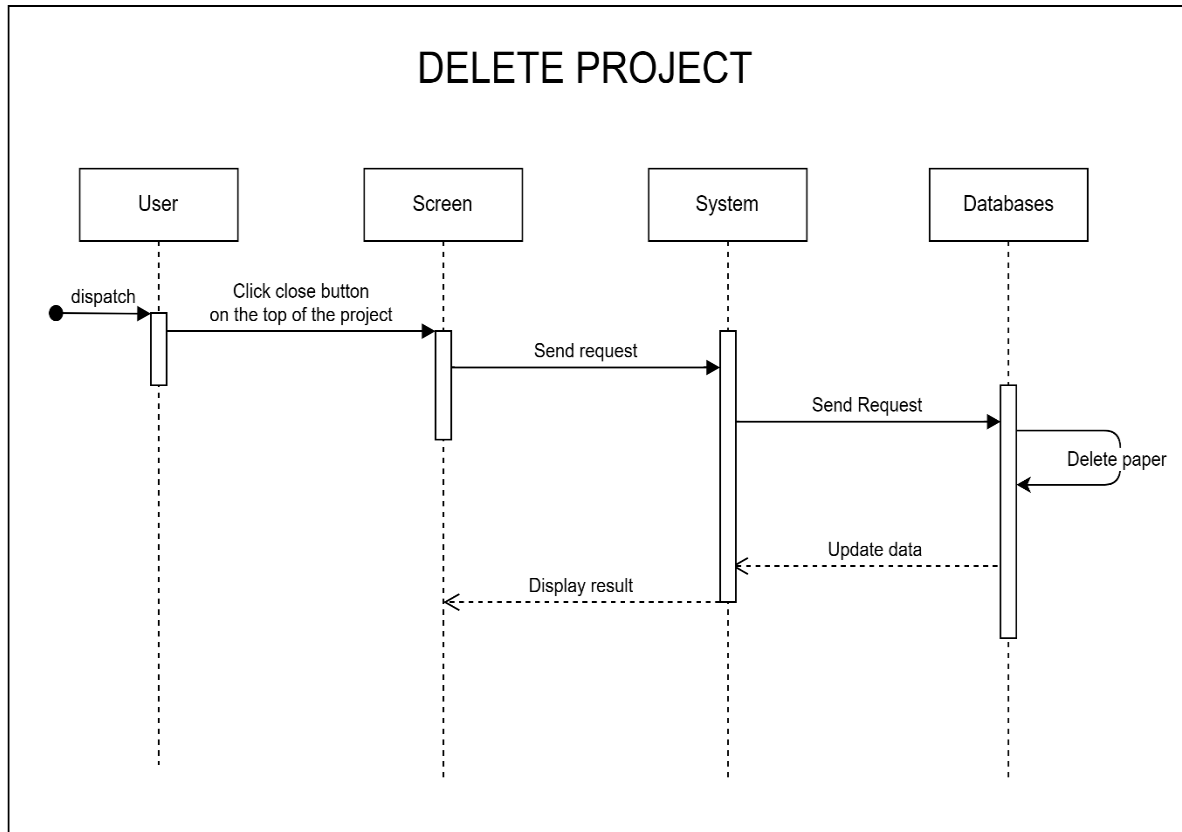
This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:

### Activity diagram



This sequence diagram will represent the interactions between different components or objects within a system and the database:

### Sequence diagram



### 3.2.6 Update project.

**ID: FR\_6**

TITLE: Update paper.

DESC: Given that a user has saved a paper, then the user must be able to update that paper.

BR:

- User clicks on the saved paper.
- Users click on function buttons to choose what to draw and draw.
- The user clicks “Save As” in the file menu or presses the key Ctrl + S.

RAT: For a user to add detail to the available papers.

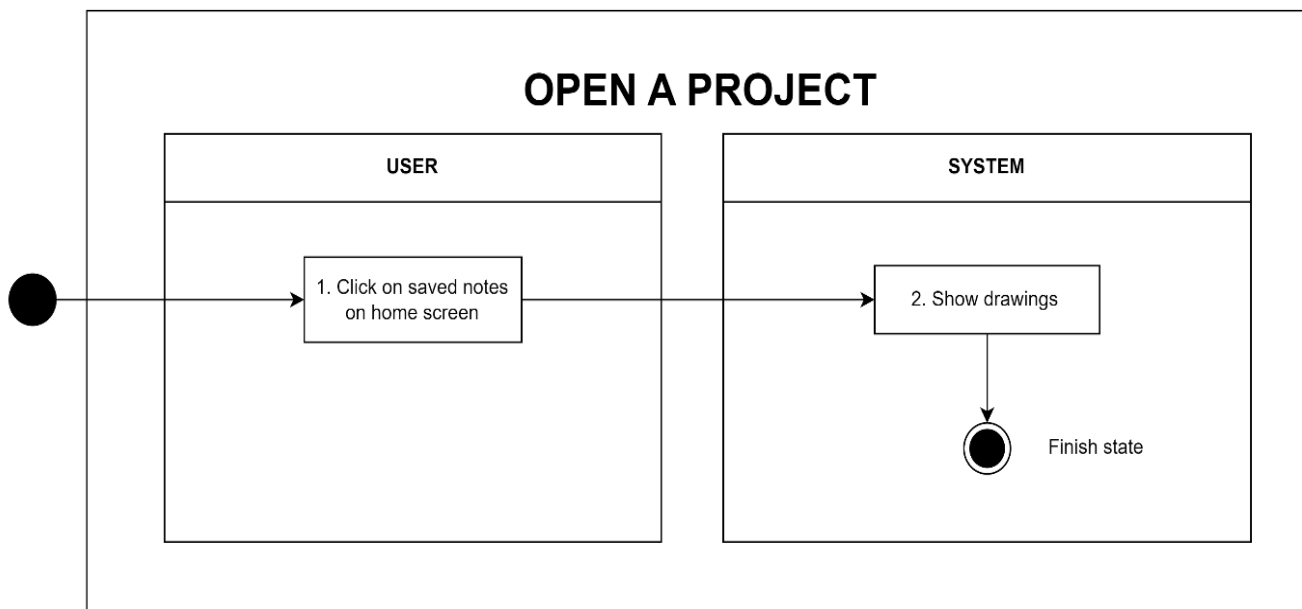
DEP: FR\_3

This is the specification of the Update project use case:

Name	Update project
Scenario	The user wants to update the project.
Description	This use case occurs when the user wants to modify the project that the user has want tp.
Actor	User
Trigger	The user logged in to the system and click to delete the project
Related use-case	
Pre-conditions	Users must be logged into the system successfully.
Post-conditions	Success: The selected project is updated from the file system. Fail: Deny request, showing error message.
Basic events	<ol style="list-style-type: none"> <li>1. Logging in to the application</li> <li>2. Choosing the file to modify then click on the “Img” button or “name.jpg” to execute modify the file</li> <li>3. The file was save/updated successfully</li> </ol>
Alternative events	
Exception conditions	<ol style="list-style-type: none"> <li>2b1. File in use</li> <li>3b1. Cannot updated successfully.</li> </ol>

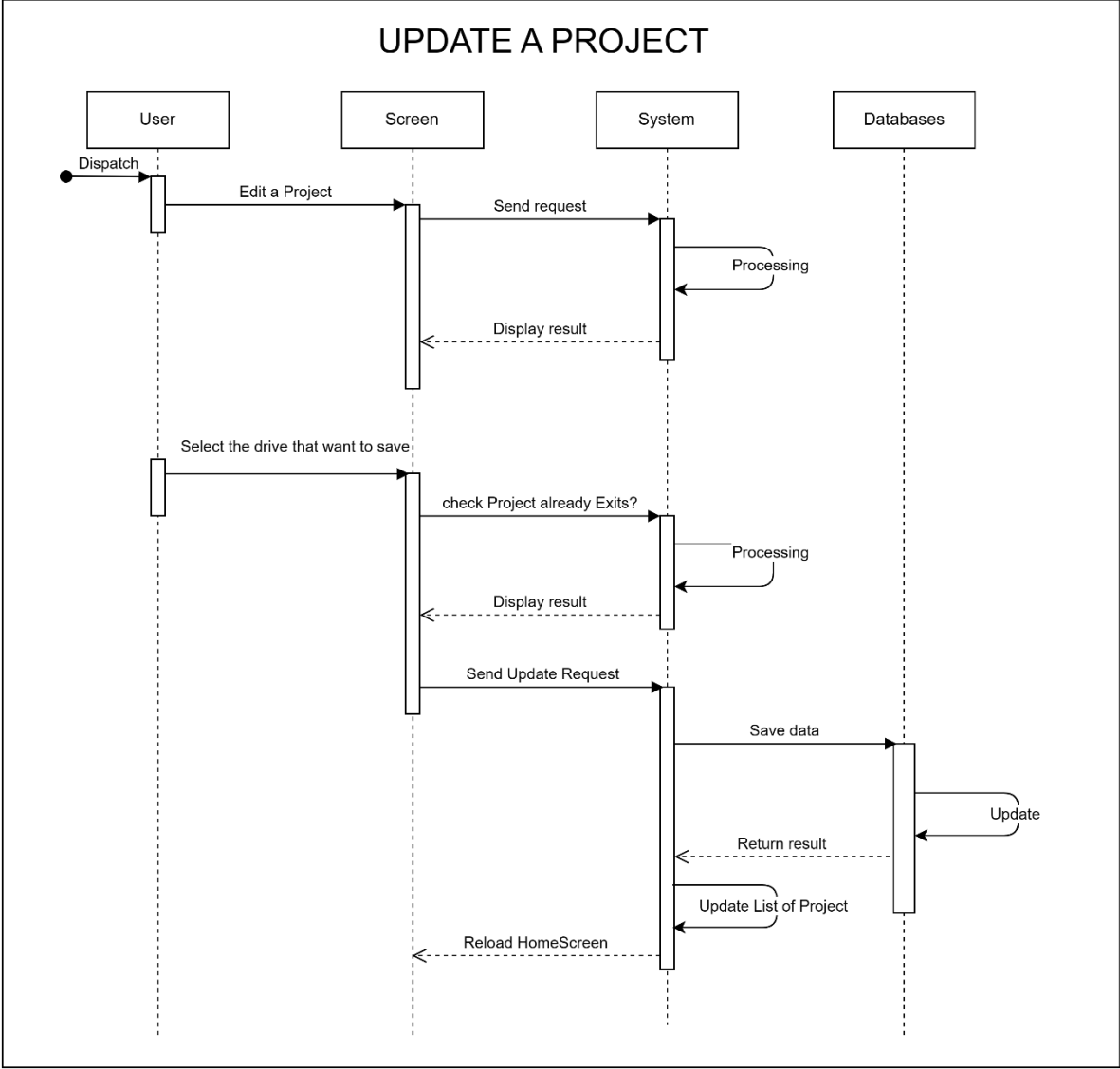
This activity diagram will model and visualize the flow of actions which helps you understand more about the scenario of the use case:

### activity diagram



This sequence diagram will represent the interactions between different components or objects within a system and the database:

### Sequence diagram



### **3.3 Nonfunctional requirements**

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

#### **3.3.1 Usability**

**ID: NF\_1.**

TITLE: Application effectiveness.

DESC: Instead of the user having to select file and open file, the application displays the drawn papers on the Homepage screen.

RAT: Users can track paper easily.

#### **3.3.2 Performance**

**ID: NF\_2.**

GIST: The fastness of saving accounts.

SCALE: The response time of saving.

METER: Measurements obtained from 10 searches during testing.

MUST: No more than 3 seconds 100% of the time.

**ID: NF\_3.**

GIST: The fastness of saving paper that has been drawn.

SCALE: The response time of saving.

METER: Measurements obtained from 10 searches during testing.

MUST: No more than 3 seconds 100% of the time.

#### **3.3.3 Security**

**ID: NF\_4.**

TAG: Communication Security.

GIST: Security of the communication between the system and server.

SCALE: The messages box shall private for log-in communications so that others cannot know the username and password from those messages box.

METER: Attempts to get username and password through obtained messages on 10 log-in sessions during testing.

**MUST:** 100% of the communication messages box between the customer and server in a log-in session should be private.

**ID:** NF\_5.

**TAG:** Login Security.

**GIST:** Security of accounts.

**SCALE:** If a user tries to log in to the web portal with a non-existing account then the user should not be logged in. The user should be notified about log-in failure: "Login failed!".

**METER:** 10 attempts to log in with a non-existing user account during testing.

**MUST:** 100% of the time.

**ID:** NF\_6.

**TAG:** Create Account Security.

**GIST:** The security of creating an account for users of the system.

**SCALE:** If a user wants to create an account and the desired user name is occupied, the user should be asked to choose a different user name.

**METER:** Measurements obtained on 10 hours of usage during testing.

**MUST:** 100% of the time.

### ***3.3.4 Audit***

**ID:** NF\_7.

**TITLE:** Ability to record performed papers.

**DESC:** Papers that have been drawn are stored in a database. Even if the user turns off the application, the data is still saved in the account.

**RAT:** The user can check papers and doesn't need to draw papers again.

### ***3.3.5 Maintainability***

**ID:** NF\_8.

**TITLE:** Application extendibility.

**DESC:** The code should be written in a way that favors the implementation of new functions to extend easily the app.

**RAT:** In the future, functions to be implemented easily in the application to develop the app.

## 4. Other requirements

### 4.1 Database

Software: Microsoft Access (Version 2000)

Purpose: The database component of Deton, as specified in this SRS, will utilize Microsoft Access version 2000 as the underlying database management system. The choice of Access 2000 is made to ensure compatibility with existing infrastructure and to leverage its capabilities in handling data storage, retrieval, and management functionalities required by the Deton application.

### 4.2 MyPaint

# Created by <https://www.toptal.com/developers/gitignore/api/csharp>

# Edit at <https://www.toptal.com/developers/gitignore?templates=csharp>

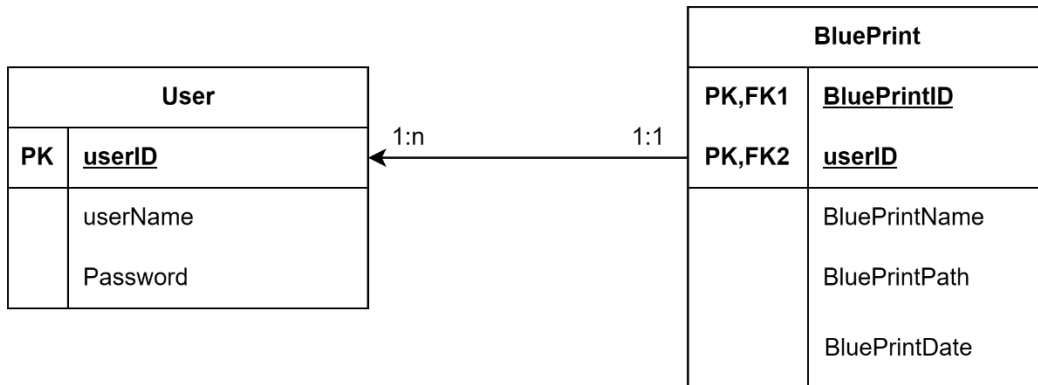
## Appendix A: Glossary

- **Query:** A database function within Deton that retrieves specific information or data from the database based on defined criteria or parameters set by the user.
- **Forms:** Interactive interfaces within Deton used for data entry, viewing, or modification, allowing users to input or manipulate information in a structured manner.
- **Reports:** Generated documents or summaries within Deton that present organized data from the database in a formatted and readable manner, aiding in analysis, decision-making, or presentation purposes.
- **Macros:** Automated sequences of actions or commands within Deton that perform repetitive tasks or operations, enhancing efficiency by executing predefined actions.
- **Validation Rules:** Specifications within Deton's database schema that enforce restrictions or criteria on data input, ensuring accuracy, consistency, and integrity of the stored information.
- **Relationships:** Defined associations or connections between different tables or entities within the database of Deton, establishing how data is related or linked together.
- **Indexing:** Optimization technique in Deton's database to improve data retrieval speed by creating indexes on specific fields, facilitating faster access to information.
- **Scalability:** The capability of Deton's database to handle increased volumes of data or users without compromising performance or functionality.



## Appendix B: Analysis Models

### ERD diagram



### Structure Diagram

