**`**

Software Requirements Specification

for

Archiving Management System

**Version 1.0**

**Prepared by Sameed Razi, Muhammad Abdullah, Ajmal Iqbal, Faiq Ahmed and Muhammad Yousuf**

**PAF KIET SE**

**25-01-2021**

**Table of Contents**

**Table of Contents [ii](#_heading=h.ihv636)**

**Revision History [ii](#_heading=h.30j0zll)**

**1. Introduction [1](#_heading=h.32hioqz)**

1.1 Purpose [1](#_heading=h.1hmsyys)

1.2 Document Conventions [1](#_heading=h.41mghml)

1.3 Intended Audience and Reading Suggestions [1](#_heading=h.2grqrue)

1.4 Product Scope [1](#_heading=h.vx1227)

1.5 References [1](#_heading=h.3fwokq0)

**2. Overall Description [2](#_heading=h.1v1yuxt)**

2.1 Product Perspective [2](#_heading=h.4f1mdlm)

2.2 Product Functions [2](#_heading=h.2u6wntf)

2.3 User Classes and Characteristics [2](#_heading=h.19c6y18)

2.4 Operating Environment [2](#_heading=h.3tbugp1)

2.5 Design and Implementation Constraints [2](#_heading=h.28h4qwu)

2.6 User Documentation [2](#_heading=h.nmf14n)

2.7 Assumptions and Dependencies [3](#_heading=h.37m2jsg)

**3. External Interface Requirements [3](#_heading=h.1mrcu09)**

3.1 User Interfaces [3](#_heading=h.2jxsxqh)

3.2 Hardware Interfaces [3](#_heading=h.46r0co2)

3.3 Software Interfaces [3](#_heading=h.2lwamvv)

3.4 Communications Interfaces [3](#_heading=h.111kx3o)

**4. System Features [4](#_heading=h.3l18frh)**

4.1 System Feature 1 [4](#_heading=h.206ipza)

4.2 System Feature 2 (and so on) [4](#_heading=h.4k668n3)

**5. Other Nonfunctional Requirements [4](#_heading=h.2zbgiuw)**

5.1 Performance Requirements [4](#_heading=h.2bn6wsx)

5.2 Safety Requirements [5](#_heading=h.1egqt2p)

5.3 Security Requirements [5](#_heading=h.3ygebqi)

5.4 Software Quality Attributes [5](#_heading=h.2dlolyb)

5.5 Business Rules [5](#_heading=h.sqyw64)

**6. Other Requirements [5](#_heading=h.3cqmetx)**

**Appendix A: Glossary [5](#_heading=h.1rvwp1q)**

**Appendix B: Analysis Models [5](#_heading=h.4bvk7pj)**

**Appendix C: To Be Determined List 6**

# Introduction

## Purpose

*This Document (SRS) lays out the description of the software that is to be developed as well as the intention of the software under development. This document also shows what the software is supposed to do as well as how it is supposed to perform, who will use it and how to use it. The document also shows how the development of the software will be done.*

## Document Conventions

*This document follows the IEEE standard. Bold faces used to emphasize section and subsection headings. Highlighting is to point out words in the glossary and italicized text used to label and recognize diagrams and tables.This Document Contains many types of fonts but to make the document more user friendly we have used bullet points and headings in bold so the client developer and users can easily understand the document there is an index page showing all the heading so the reader can easily jump to his requirement.*

## Intended Audience and Reading Suggestions

*This document can be read by all those who are developing or testing and is for those who are even using this it is a total guide line to those mentioned people.*

## Product Scope

*The Tracking of document manually is waste of time so we have developed a software that can store electronic documents in form of files and also keep track of the physical document itself that where it is held so searching it would take few seconds to get to it .The feature to retrieve document online is the main reason you want to set this software up for your organization.*

## References

*This product is just a modern way to archive documents and files in a manner that retrieving and searching could be done easily; it is a new product no reference is taken from anywhere.*

# Overall Description

## Product Perspective

*The AMS project is a new, self-contained system intended for web application. This system is intended to replace the manual paper-based storing to digital, high quality, cost effective and reliable flow of documents and files through-out the organization. The AMS system to be developed benefits greatly the employee’s as well as their supervisor. The other intent of this system is to guide users to easily track and get to a document without any fatigue or hard work. Web apps serve as the frontend, AMS backend responsible for database and information transaction, CRUD services.*

## Product Functions

*1. User Registration 5. Document Registration*

*2. Department Registration 6. Document Searching*

*3. Rack Registration 7. Document Uploading*

*4. Box Registration 8. Document Encryption*

## User Classes and Characteristics

*So there are 5 types of users.*

*The person who has all the functionalities is called admin or superuser.*

*The person who can only insert a new document in the system is called the document adder.*

*The person who can receive a document or dispatch a document is called the document controller.*

*The person who can see all the documents in all the departments is known as the document in all departments .*

*The person who can see in only the documents in his department is known as the document in the department.*

## Operating Environment

*The developed Application should run under any platform that contains a web browser and has internet connection on it.The admin and all users must have access to the internet and have their personal computers with them to run this web application. If the user which adds a new document needs to add a new department etc must first forward that to the admin as he can only do that other users don't have access rights.each user has access levels which determine what privilege they have in seeing or getting the document.*

## Design and Implementation Constraints

*If the user inserts a file onto an already inserted file then the previous file can not be retrieved.*

*Users can have many kinds of contacts.*

*Users can have one or more roles.*

## User Documentation

*The below link takes you to a tutorial “how to use the software”*

[*https://www.youtube.com/watch?v=WYT3eGHxXaw*](https://www.youtube.com/watch?v=WYT3eGHxXaw)

## Assumptions and Dependencies

1. *Each user will have unique login credentials.*
2. *There will be a backup database for sensitive information so it cannot be lost.*
3. *Assuming all users which use this have the basic computer and english language knowledge.*

# External Interface Requirements

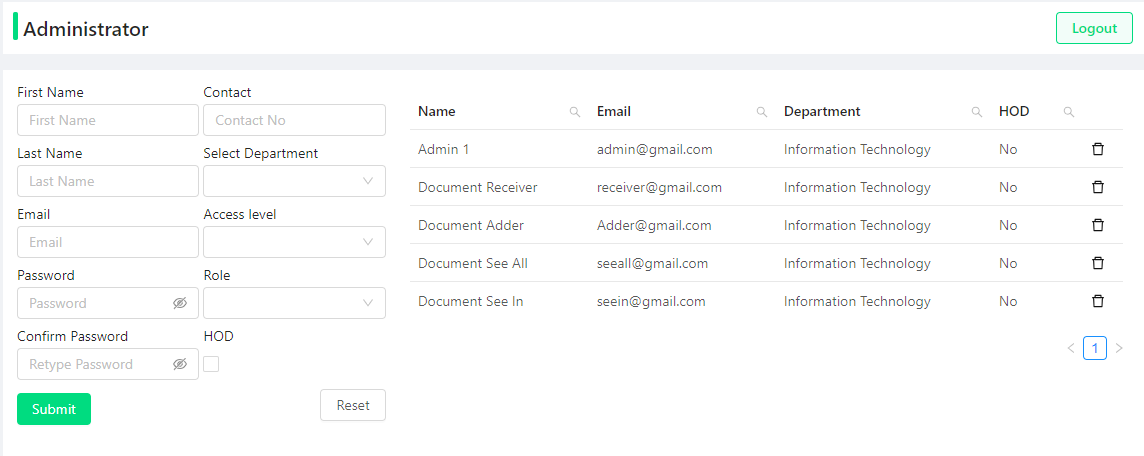
## User Interfaces

*The User Interface looks like this.link to a tutorial is given below “how to use the software”.*

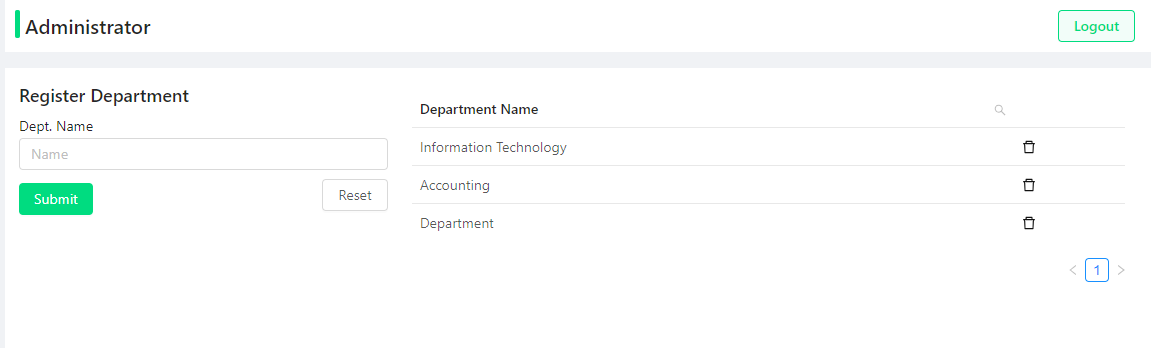
*when user first approaches to sign in he see this page :*

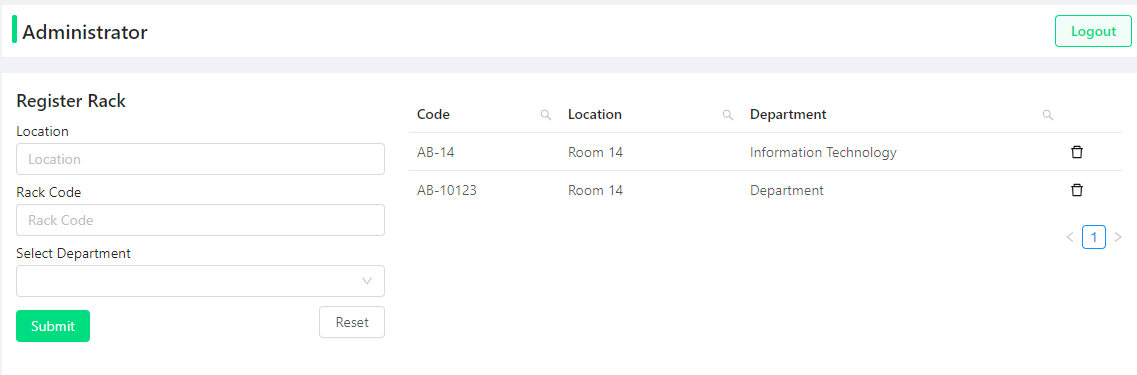
**

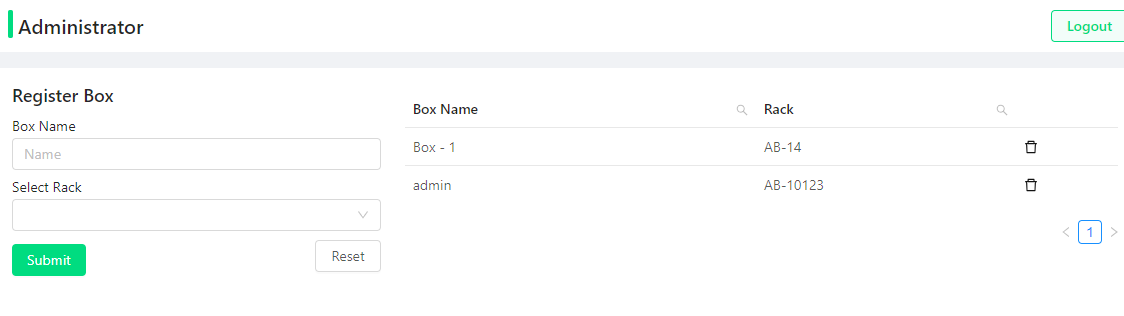
*When the Admin is inserting a new user in the system :*

**

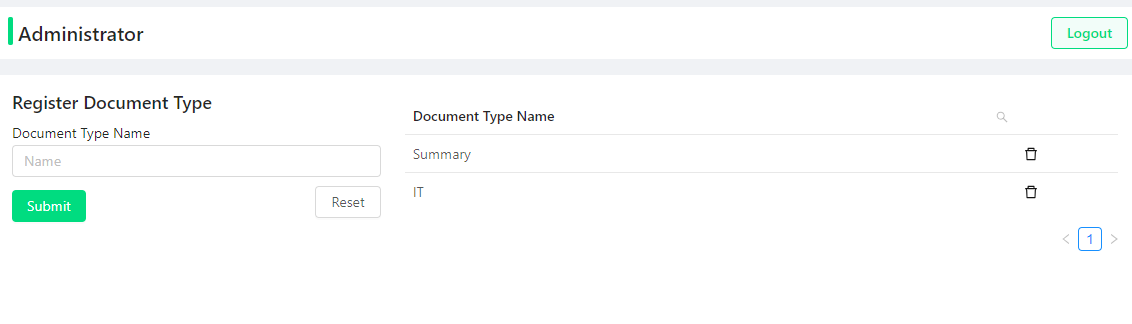
*When the Admin is adding a department with a rack and box in it :*

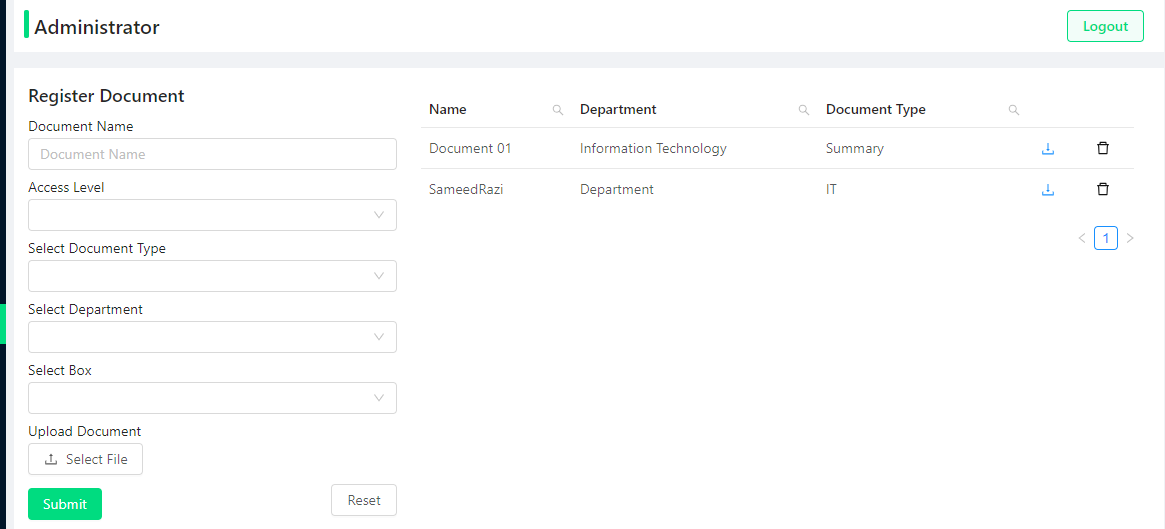
**

**

**

*When the Document is inserted :*

**

**

## Hardware Interfaces

*This software can run on any operating system that has a browser which supports javascript html and css with internet connection.*

## Software Interfaces

*This web application is in three parts which is the frontend backend and the database*

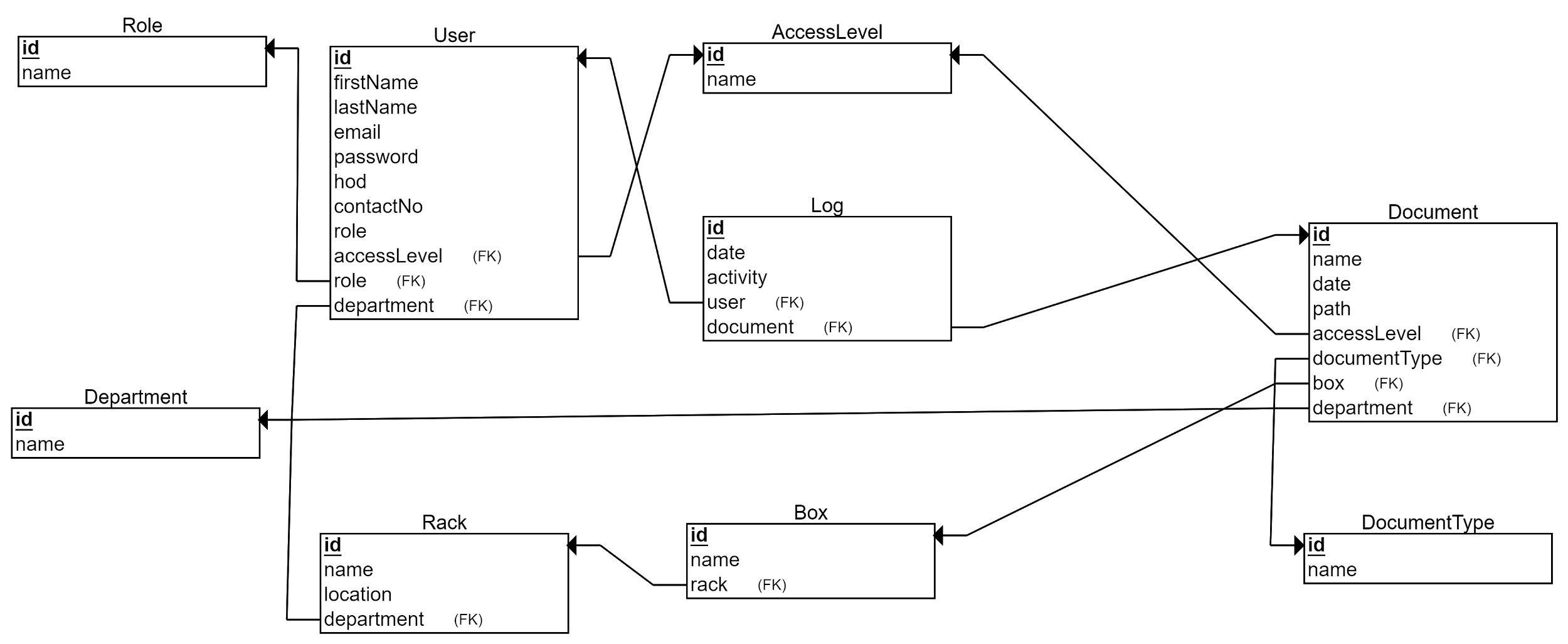
1. *The front end runs on the latest version of react framework with few third party libraries.*
2. *The back end runs spring boot server which is built on maven dependencies.*
3. *The database is a sql database called mysql database.*

## Communications Interfaces

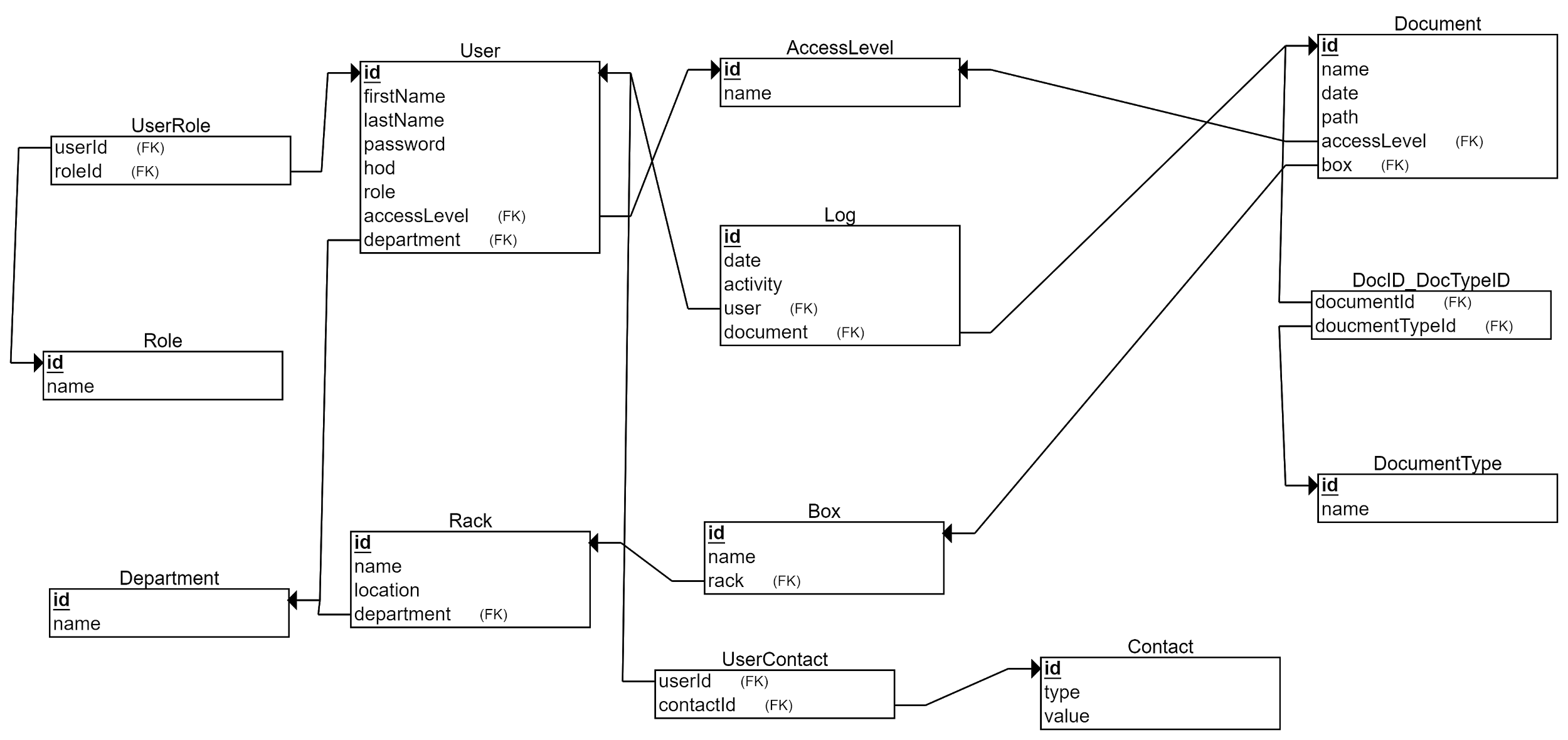
*The communication between the user and the server is secure as we are using spring security which needs some kind of token from the client which is done automatically using cookies.the documents which are stored like a file system are encrypted so a secret key word can only decrypt it.*

# System Features

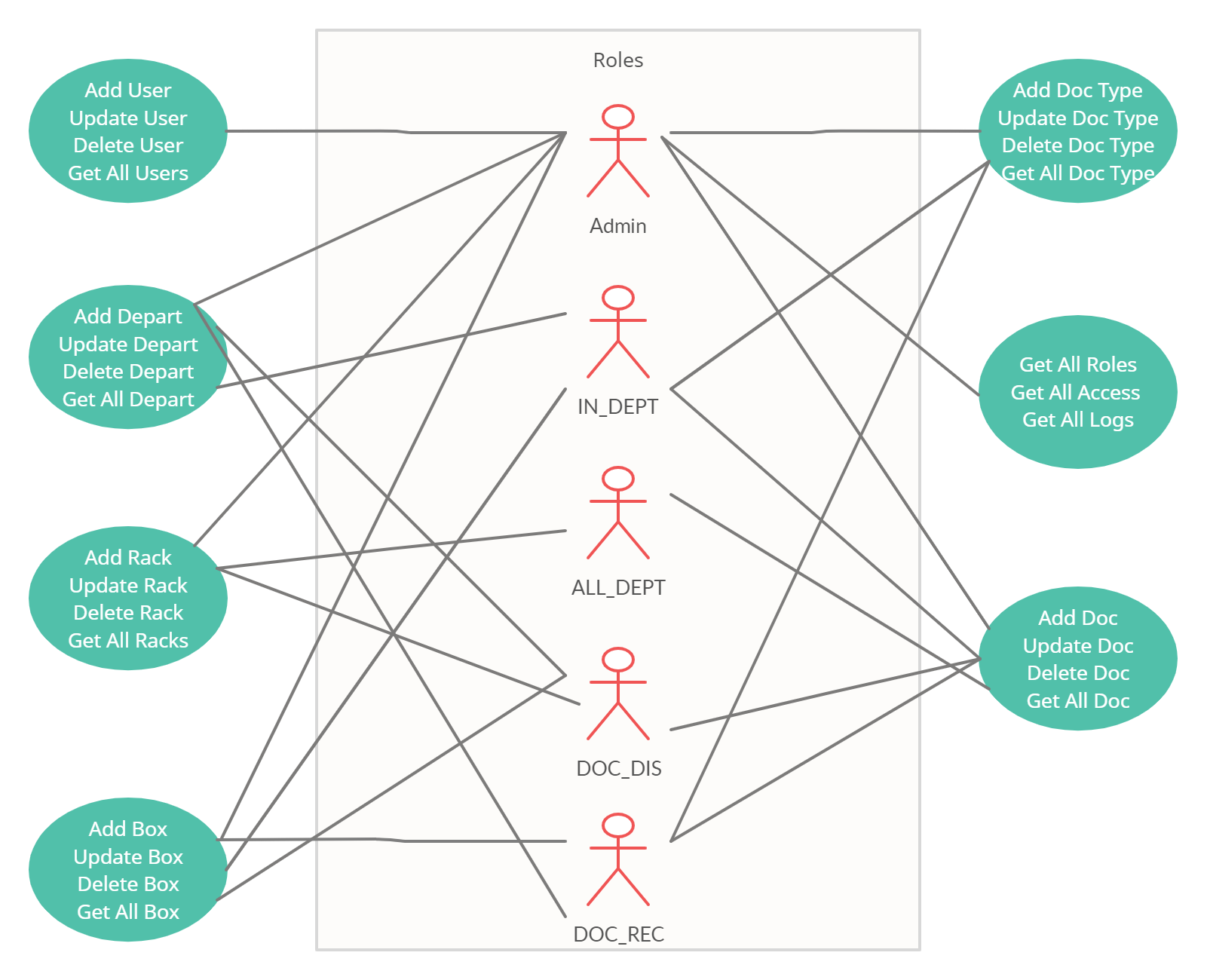
*The Applications Entity Relational Diagram :*

**

*The Application Enhanced Entity Relational Diagram :*

**

*The Applications Use Case Diagram:*



## User Registration

4.1.1 Description and Priority

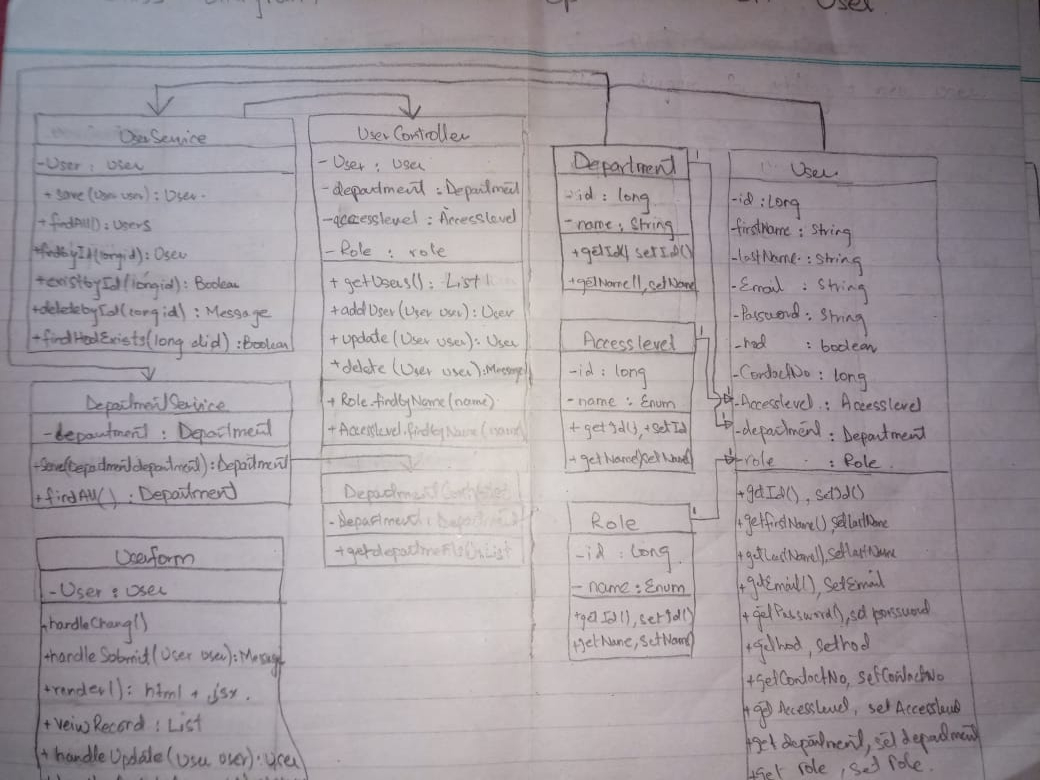
*The admin first has to make users according to their role and access level in the organization.*

*It is important to make users because they are the ones using the web application.*

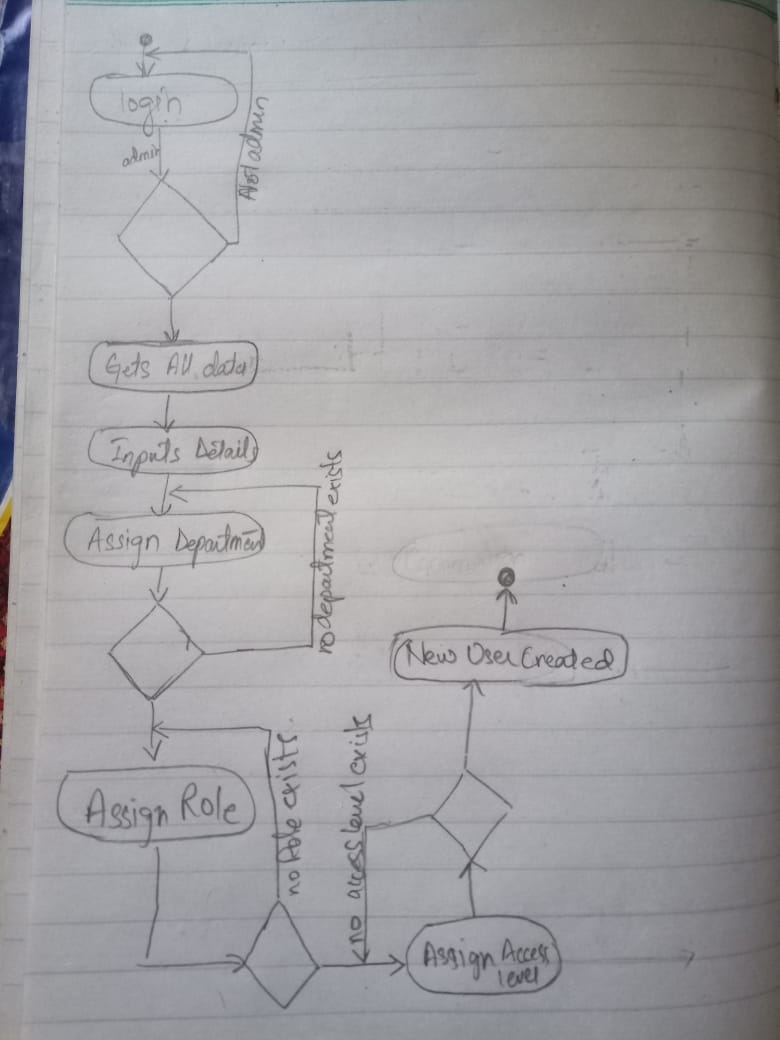
|  |  |
| --- | --- |
| ***ID:*** | ***4.1*** |
| ***Title:*** | *User Registration* |
| ***Description:*** | *New user register according to there access level and roles* |
| ***Primary Actor:*** | *Admin* |
| ***Preconditions:*** | *Admin has to login in the system to do this.* |
| ***Postconditions:*** | *A new user will be inserted in the database and now he can login to do his respective jobs* |
| ***Main***  ***Success Scenario:*** | *The admin logged in the system.The system checks if this is an admin,now he can view the registration page he has to type in the user details and select their privileges.if there are no validations then the system will insert the new user in the database.* |
| ***Extensions:*** | *if the user is not an admin he cannot see the registration page, if the email inserted in the details already exists in the database the system will generate an error,if there is already a head of department of that department new user cannot be a hod.* |
| ***Frequency of Use:*** | *This transaction will be used less.* |
| ***Status:*** | *It has been developed* |
| ***Owner:*** | *Group Bytes Kiet* |

4.1.2 Stimulus/Response Sequences

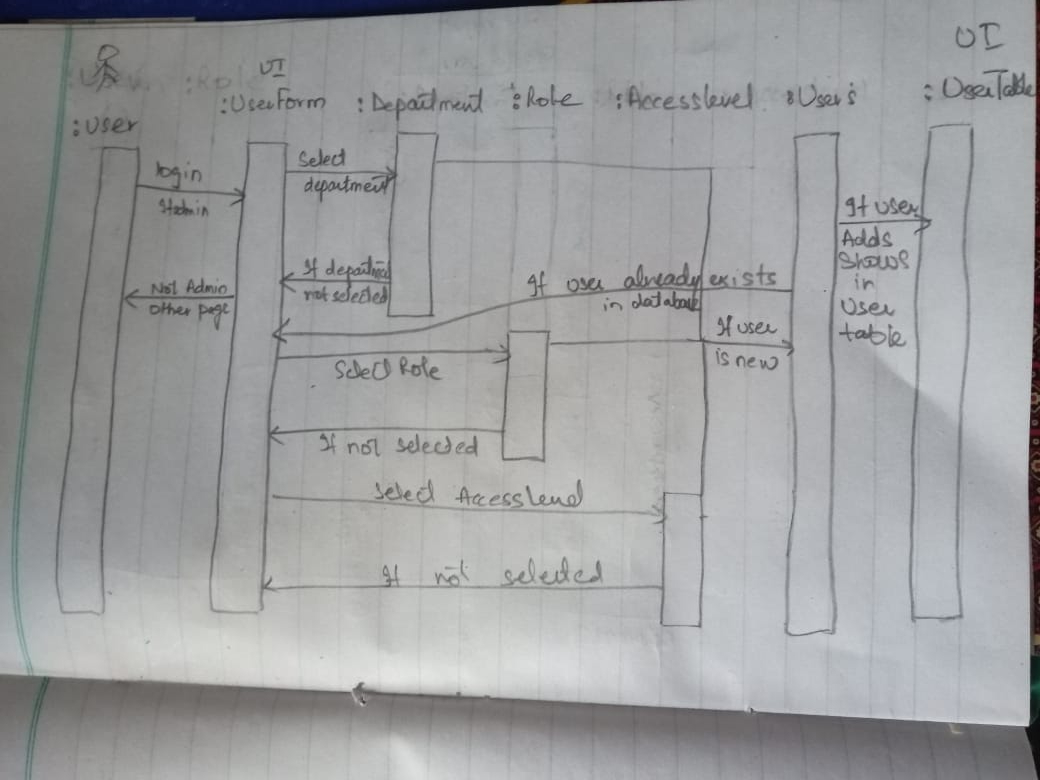
*User Transactions Class Diagram :*

**

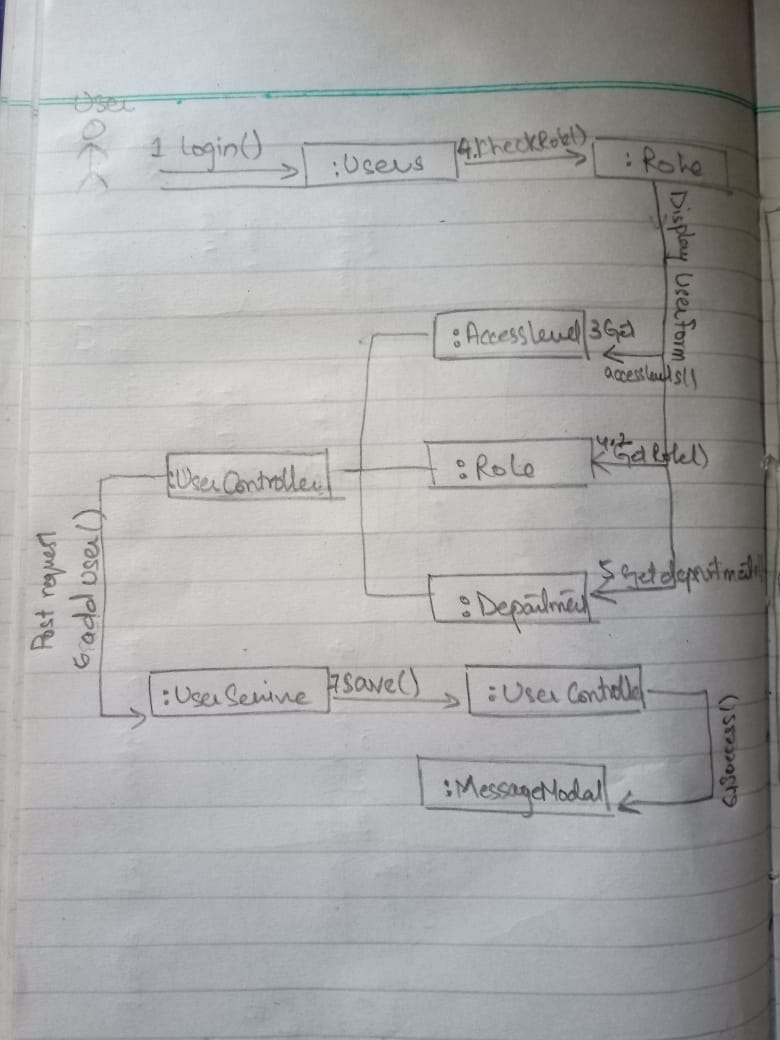
*User Transactions Activity Diagram :*



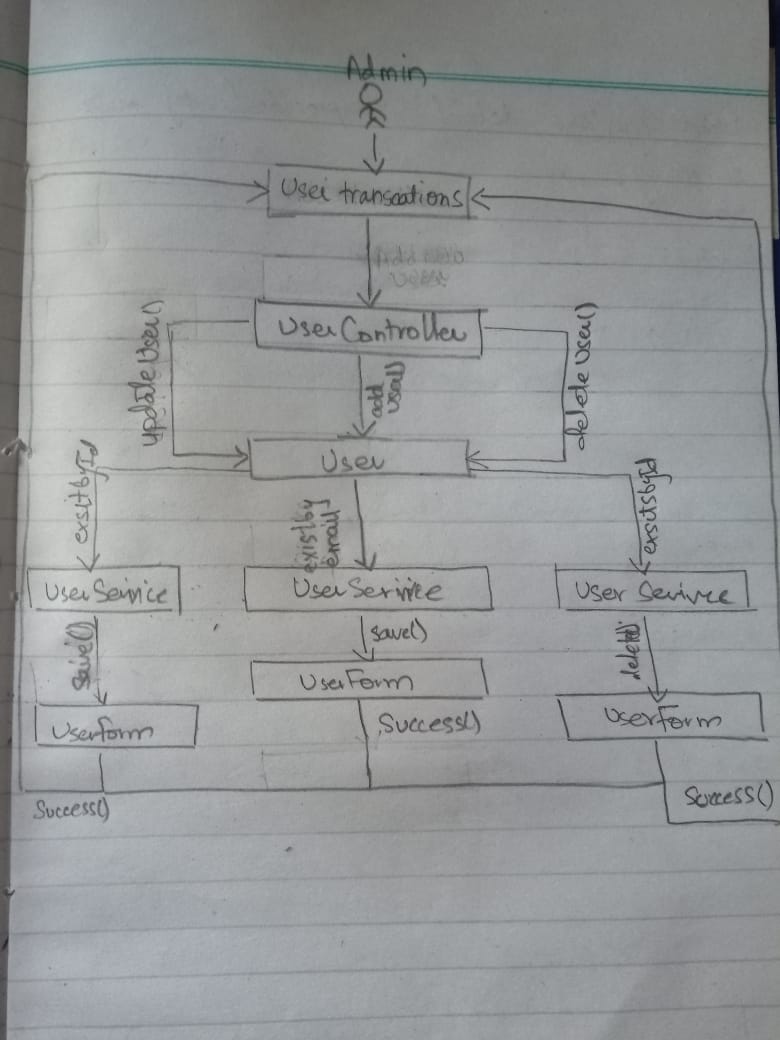
*User Transactions Sequence Diagram :*

**

*User Transactions Collaboration diagram :*

**

*User Transactions State Chart Diagram :*

**

# Other Nonfunctional Requirements

## Performance Requirements

*The archive should provide the ability to handle (moderate, high, or very high – whichever applies to you) volumes of ingestion within time windows needed to provide the business with access to documents when needed within business processes. In addition, the archive should provide reasonable response times for document search and retrieval, and the solution should have the ability to perform ingestion and archive functions without negatively impacting overall system performance for users.*

## Retention and integrity

## *The archive should be able to retain documents for defined periods of time, taking into account legal, regulatory, fiscal, operational, and historical requirements. In addition, the archive should provide a suitable guarantee of authenticity. Finally, (if this applies to you) the archive should provide the ability to retain information on unalterable storage platforms when needed (e.g. WORM storage for SEC 17a-4 compliance).*

## Security Requirements

*The archive should have the ability to restrict access to documents, such as for documents that are private, confidential, privileged, secret, or essential to business continuity. This may include requirements for encryption of stored content.*

## Accessibility and availability

*The archive should provide a mechanism for authorized users to search for and retrieve documents. In addition, the archive should provide the ability for certain external users to retrieve documents, such as e-presentment for customers or agents.*

## Business Rules

***1.******Admin***

*He can add new users.*

*He can add new departments.*

*He can add new racks.*

*He can add new boxes.*

*He can add new documents.*

***2.******Document reader in department***

*He can only see documents according to his access level only in his department.*

***3.******Document reader all department***

*He can see documents of all departments according to his access level.*

***4.******Document Receiver***

*He can receive documents and archive them back where they belong.*

***5.*** ***Document Adder***

*He can add new documents into the system*

* 1. **Other Requirements**

We are using mysql database and to ensure that no other types of roles or access levels can be inserted in the database we have to run some queries when initializing the database for the first time.