



**spring2021.cse327.1.9 – Project Final Report**  
**Document Digitization**  
**Dr. Nabeel Mohammed (NBM)**

**Group Details:**

<b>Name</b>	<b>ID</b>	<b>NSU Email</b>
Md. Sharukh Fardin	1821463042	sharukh.fardin@northsouth.edu
Sajib Hossain	1821245642	sajib.hossain03@northsouth.edu
Md Kamrus Samad	1813059642	kamrus.samad@northsouth.edu
Tanvir Al Mahmud	1813564042	tanvir.mahmud15@northsouth.edu

**Work distribution among members:**

<b>Topic</b>	<b>Members</b>
Web Version	Sharukh, Samad
Android Version	Sajib, Tanvir
Project Planning	Whole team contributed equally
SRS, SDS	Whole team contributed equally
Project Report	Whole team contributed equally

## Missing Features:

### Web:

1. Tried to implement google login according to the project description. But there was an issue. We tried to create an internal organization website in this project. If we had implemented the google login feature, any google user with an account could visit our website and then login. That's why we removed the work we did on google login API.
2. We had plans on messaging section for our project. But in the end, we could not implement it because of the lack of time.
3. Some features are not as in depth as we imagined it to make.

- Sharukh and Samad

### Android:

We were working simultaneously on android studio. As it was our first project on android, we were facing a lot of problem working together. One of our machines in which we worked on missing android emulator. We have to test our project on single machine remotely. We have designed all our frontend and test it working on local machine but were unable to connect it to the same backend through API for web application.

- Sajib and Tanvir



# Software Requirements Specification For 'DOCUMENT DIGITIZATION'

Prepared By:

Name	ID
Sajib Hossain	1821245642
Md.Sharukh Fardin	1821463042
Md Kamrus Samad	1813059642
Tanvir Al Mahmud	1813564042



# Table of Content

<b>1. Introduction</b>	<b>3</b>
- Purpose	
- Document Conventions	
- Intended Audience and reading suggestions	
- Product Scope	
- References	
<b>2. Overall Description</b>	<b>4</b>
- Product Perspective	
- Product Functions	
- User Classes and Characteristics	
- Operating Environment	
- Design and Implementation Constraints	
- User Documentation	
<b>3. External Interface Requirements</b>	<b>6</b>
- User Interfaces	
- Hardware Interfaces	
- Software Interfaces	
- Communications Interfaces	
<b>4. System Features</b>	<b>7</b>
- Records Management	
- Accounts Management	
- Search and Retrieval Management	
- Reports Management	
- Notification Management	
- System Management	
- Document Digitization Services	
<b>5. Other NonFunctional Requirements</b>	<b>10</b>
- Performance Requirements	
- Safety Requirements - Security Requirements	
- Software Quality Attributes	
- Notification Management	
- System Management	
- Document Digitization Services	
<b>Appendix A: Glossary</b>	<b>11</b>
<b>Appendix B: Analysis Models</b>	<b>12</b>



# Introduction

## Purpose:

The purpose of this document is to describe and represent document digitization system "DOCUMENT DIGITIZATION SERVICE" in detailed form and all its functionality as well as how it will work. This SRS documents all segments of the aforementioned "DOCUMENT DIGITIZATION SERVICE" system.

## Document Conventions:

Standard typographical convention will be held during the project work and IEEE standard will be maintained for this document. All the color and font will be used and adjusted in user-friendly way so that client can easily grasp and understand the complexities and hierarchy for all functions that have been used.

## Intended Audience and Reading Suggestions:

This document is aimed for the design analysis team who will involve the developer team with the requirement of the project and make as well as describe all the requirement as clear and detailed as possible.

## Product Scope:

The web and android app we are developing for cross-platform use is an easily maintainable for developer and capable of handling large scale of data for user. User will be able to track all data. Prioritized user will be able to track down other user activities and customize stored data and user right as needed. This product is not totally self dependent this will use some prebuilt method and third party API for some device to smooth user experience and optimize system performance.

## References:

SRS IEEE standard format.



# Overall Description

## Product Perspective:

The perspective of this product is to create a document digitization and management platform for employees with necessary admin controls. The product will support multiple target platform (Android & Web).

## Product Functions:

The product will have three groups of actors. They are:

1. Administrator
2. Manager
3. Employee

The functionality of the product includes:

- Records Management
- Accounts Management
- Search and Retrival Management
- Reports Management
- Notification Management
- System Management
- Document digitization services

## User Classes and Characteristics:

This product will include three user classes:

- Administrator
- Manager
- Employee



## Operating Environment:

The operating environments of this product are:

- Operating System: Windows(Computer), Android(Mobile Phone)
- Hardware Platform: Personal Computer(PC) and Mobile Phone
- Distributed Database
- Database: SQLite

## Design and Implementation Constraints:

Constraints include:

- The documents should not be handwritten.
- The supported operating system will be windows and android
- SQL automatic commands calling with the help of API

## User Documentation:

Users will be able to view dashboard where all functions(features) of the product will be visible.

## Assumptions and Dependencies:

The OCR and other libraries used in this system can have impact on it. Besides python sometimes is not backward compatible in updates.



# External Interface Requirements

## User Interfaces

### WebApp:

- Front-end : HTML5, CSS3, Bootstrap, JavaScript.
- Back-end : Python, SQLite, Django framework.

### Mobile App:

- Java

## Hardware Interfaces

- Personal Computer
- Android Mobile

## Software Interfaces

- Since most of the users are android user so we have picked the Android as the primary OS
- User can access using web browser also for PC
- To interact with the database, we'll be using Django.
- Django will be used to interact with the centralized server.
- In order to save the records SQLite database will be used.

## Communications Interfaces

This software will communicate with google login through API for login verification. Will also communicate with sqlite server for data store and update. For notify user through email about any kind of change in database system it will use API connecting google email service with the system.





# System Features

## Records Management:

- Convert original paper into digital format and integrate with the existing document without using any third-party software.
- The System should have dashboard for private and public folders, uploaded documents and notifications.
- The System should have no limit for creating folder or subfolder and should have a public repository where users can only view the uploaded/created record in assigned public folder as well as a private repository where only permitted users and groups are allowed to view and edit a record.
- The System must be able to lock and unlock folder and capable to view the previous document uploaded
- The system should be capable to assign user and groups to specific folder/files and capable to upload a single or bulk scanned document.
- The System will have the feature for indexing each record and should have no limit in index field.
- Should have a field to upload date and expiry date for archiving purposes and will automatically archive and capable to set the parameters for archiving.
- Capability for uploading index file in csv format and automatically link the attributes designated file names without any limit.
- The system Capture specific index in uploaded documents and automatically fill up the designated field.
- The system should be capable o route records to one or more user accounts.

## Accounts Management:

- The System should have unlimited creation of user accounts.
- There will be a password reset tool for the end user. The system will notify the administrator via mail when a password change request will be made.
- The system will have user account link and avatar upload feature for the user.
- The system will give the administrator the user management option so that he can create new user accounts with specific roles such as employee, manager and administrator.
- The system will have a option for the manager to create group of employees.
- The system will have Google login authentication option.



## Search and Retrival Management:

- There will be search option in the system to search data such as index, file name, date, author name, uploader name.
- The system will have an advance content search that can search optical character recognition (ocr) of the scanned documents.
- The system will have search filtering options.

## Reports Management:

- There will have counting option in the system to count uploaded records by folder or user and all pages of every record.

## Notification Management:

- The system should send notifications for every routed task. Employee's tasks to manager and manager's task to the administrator.
- The dashboard will consist option to view real time notification.

## System Management:

- The system can be accessible in web browser and can support multiple platforms such as Internet Explorer, Google Chrome, Mozilla Firefox.
- The system have open source programming language using a Service Oriented Architecture (SOA).
- The system will be in django and responsive view from desktop, tablet and mobile.
- The system should have policy such as accounts with 3 failed login attempts should block and Reset of password should be done by the assigned administrator, idle time of logout is automatically 15 minutes, password and username should be in alpha numeric keys.
- The digitized document must be searchable and Shall display search output and can be linked into the records on File Server.
- For the digitization project, the service provider also had highly technical staff.
- To ensure continuity of service, maintenance, and proper technology transfer, provide advanced skills for the technical staff who will be assigned by Plan International Bangladesh.
- The system must have a backup and recovery mechanism for the digitized documents.



## Document digitization services:

- Scanning, indexing, unfastening and refastening (in cases of staple wires, clips and etc.) of documents will be done onsite to be determine by the Plan international Bangladesh, with an 8 hrs. x 5 days a week provision of manpower, high speed scanner and back-up facilities.
- There will be three index per document, index data to be entered in placed in a separate paper, positioned on the top of each document. Sizes of document will be scanned but not limited to the following: A4, Short, Legal or A3. Scanning of documents will be done onsite to be determined by the implementing agencies.
- Scanned images will be saved using industry standards, such as, TIFF, G4,PDF/A, searchable PDF or a like. Where minimum resolution will be 20dpi.
- Digital Images can be viewed and printed using standard PC and Printer.
- Uploading of digitized records into a defined storage area.



# Other Non-Functional Requirements

## Performance Requirements

For this system one of our main concern is performance. Since the core objective of this project is to digitize existing documents/files from the Department of Finance, we have to prioritize time.

## Safety Requirements

We won't have to deal with security problems too much because this is an online platform. However, in order to prevent unnecessary risks and emergencies, the system would be able to warn other users who are breaking community standards and in case of emergency it will suspend the user immediately and in case of system failure automatically system data will be backed up.

## Security Requirements

Users will be able to use their Google accounts to register. This will establish whether or not they are genuine users. In case of any kind of data and account information change administrator will be notified as early as possible and give limit for login attempt to avoid brute-force attack.

## Software Quality Attributes

### Adaptability

We'll try to make it as adaptable as possible to fix problems.

### Availability

The system will remain within the database and there will be no time constraints. It will be also portable as user can access the software through android devices and Internet services.

### Maintainability

To improve its reusability and efficiency, the platform will be maintainable efficiently and method should be written in such way that other developer could work and maintain as per need very easily.



## Appendix A: Glossary

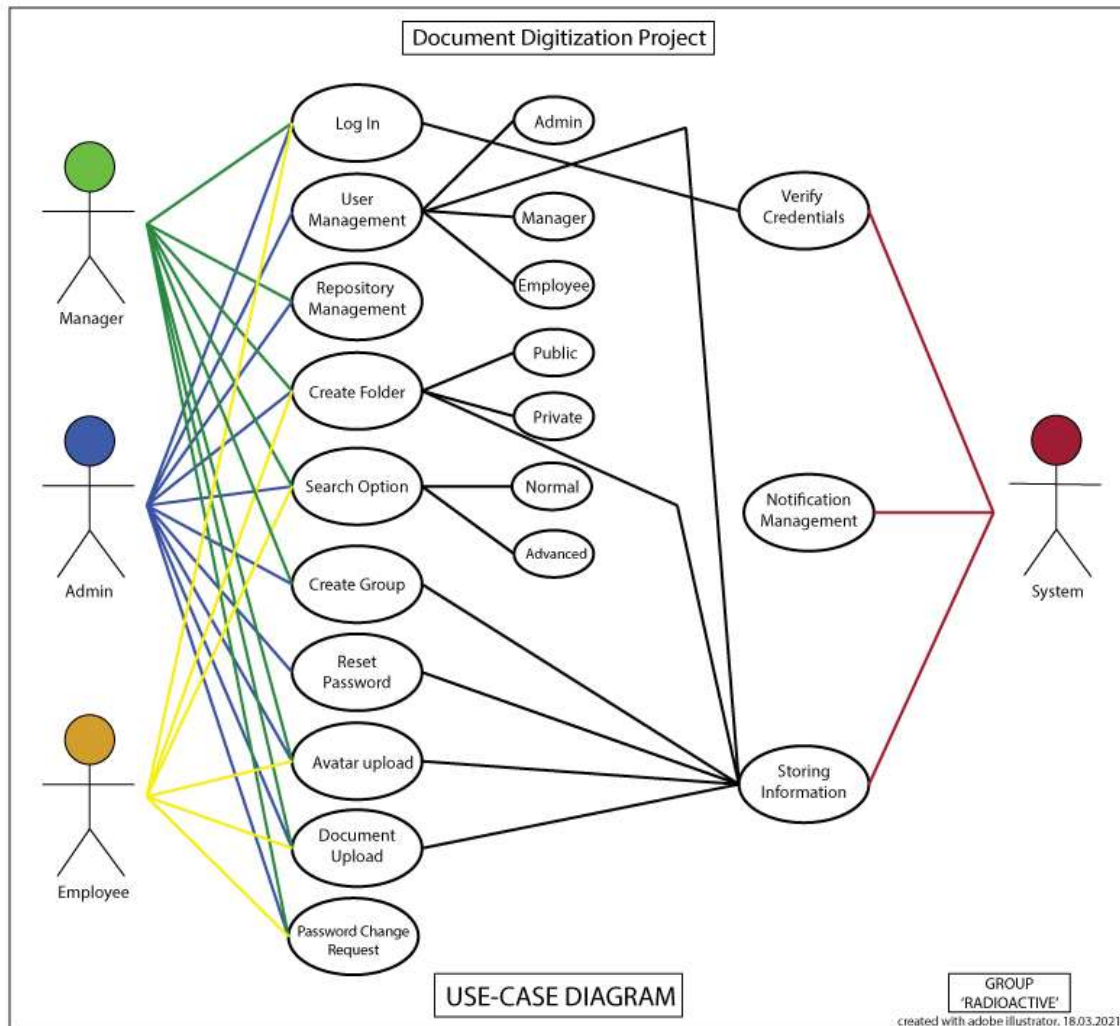
Repository : a central place where data is stored and maintained; it can be either remote or local to the user or system

OCR: Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo.



## Appendix B: Analysis Models

### Use-Case Diagram:





## Expanded Use-Case Diagram:

Actor Action	System Response
1. Go to the website.	2. Login page served.
3. Input login information and press login button	4. Verify credentials. If user is found then redirect him to user dashboard. Otherwise give error message and block user account on 5 failed login attempt.
5. User sees many options in dashboard such as create folder, upload document, password reset request, user profile information with avatar upload feature, normal search option for document and users and many more stated in the use-case diagram above.	6. System will serve according to user request and permission. System will also send notification to appropriate user for specific requests.
8. Advanced search option will be available that will support OCR converted text.	7. System will input all the added and changed data to the database after user has performed action.
10. If done, user will press log out button.	9. Show result of search result from database. The result will have sorting option.
	11. Log out user.



# Software Design Specification For 'DOCUMENT DIGITIZATION'

Prepared By:

Name	ID
Sajib Hossain	1821245642
Md. Sharukh Fardin	1821463042
Md Kamrus Samad	1813059642
Tanvir Al Mahmud	1813564042





# Table of Contents

1. Introduction.....	2
2. Document Conventions.....	3
3. Description of Design Components.....	3
4. High Level Component Design.....	4
5. UML Class Diagram With Description.....	5
6. Sequence Diagrams.....	11
7. Database Design.....	12
8. Implementation Plan.....	14

## Introduction

### Purpose

This document will define the Software Design Specification for "DOCUMENT DIGITIZATION". It contains specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document. This document will outline the software design and specification of our workflow task management system in addition to system architecture, system components, and software requirements. This document also specify and visualize all-requirement by client, hardware specification, software architecture and database system for the software

### Scope

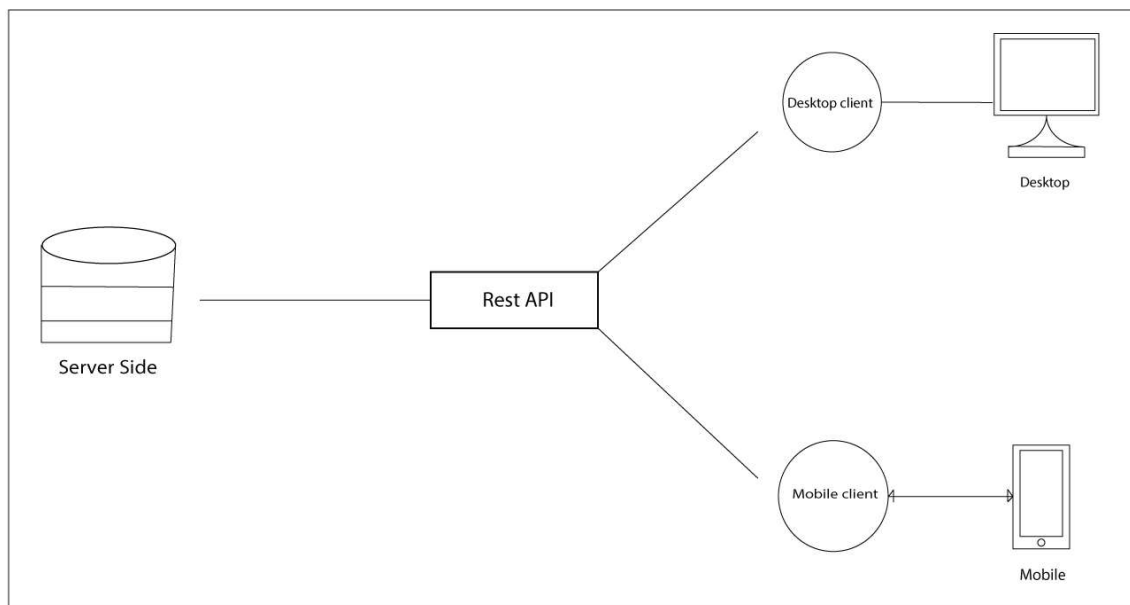
This Design Specification is to be used by Team Radioactive as a definition of the design to be used to implement the software system for data digitization which will enable user to upload, update and customize document with specific role and right.



# Document Conventions

The IEEE standard will be maintained for this document with the standard typographical and design conventions.

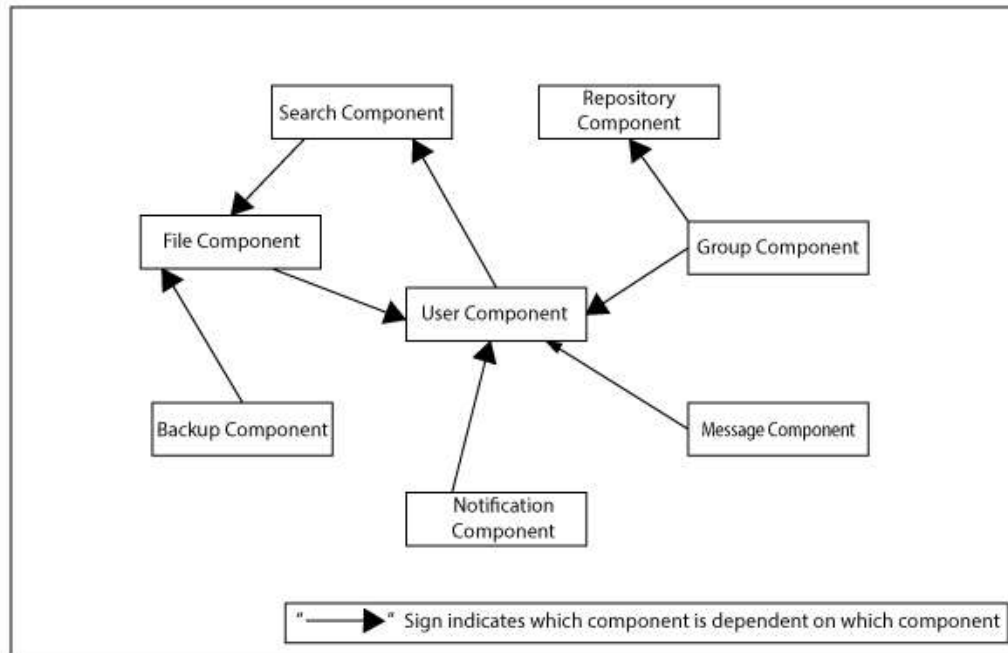
## Description of Design Components



There are 4 total components. They are Server side, API(REST API will be used) and client side. In client side, there are Desktop and Mobile client. Server performs the operations in a client-server relationship in a computer network. Server will store and process all the data's that are used. In client side, the view will be generated. The REST API will connect the server with two of the clients.



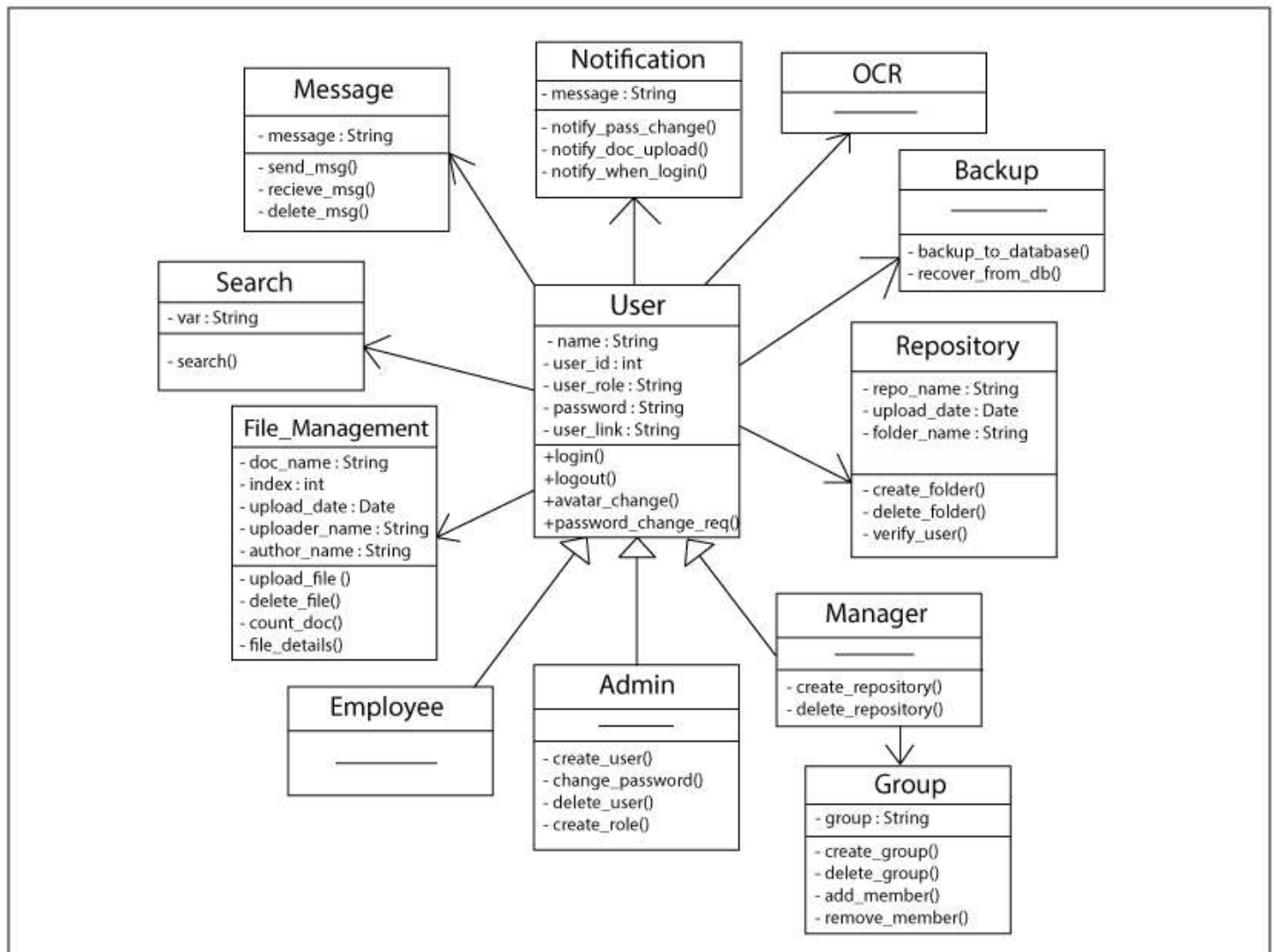
## High Level Component Design





# Class Diagrams and Description

## UML Class Diagram





## Detailed Class Description:

### Class: File Management

Purpose of this class is dealing document file details.

#### Attributes:

- doc\_name (String): This attribute takes the document file's individual name.
- index (Int): Counting the uploaded file.
- upload\_date (Date): Storing the document uploading moment.
- uploader\_name (String): Who uploads the document.
- author\_name (String): Writer of the document.

#### Methods:

upload\_file (doc\_name, index, upload\_date, uploader\_name, author\_name)

- Return type: Void
- Purpose and description: Uploading document file send to server by user.

Delete\_file (doc\_name)

- Return type: Void
- Purpose and description: Deleting document file from server by user.

Count\_doc (index)

- Return type: Int
- Purpose and description: Counting how many documents are stored

Files\_details ( )

- Return type: String
- Purpose and description: Delivering the details about file (uploader name, date etc.).

### Class: USER

Purpose of this class is dealing user details.


#### Attributes:

- name (String): This attribute takes the user name.
- user\_id (Int): Every user will have individual id.
- user\_role (String): Position in the company.
- password (String): User password for secure his account.
- user\_link (String): ??????

#### Methods:

login (user\_id, password)

- Return type: Void
- Purpose and description: User can login and landing to the dashboard.

logout ( 

[Print](#)

- Return type: Void
- Purpose and description: User can logout from account.

Avatar\_change ( )

- Return type: Int
- Purpose and description: User can use or delete or change avater.

Password\_change\_req ( )

- Return type: Void
- Purpose and description: User can send request to admin for changing his password.

## Class: Employee

This class is a child class of User class. This class can handle employee's purpose and documents.

## Class: Admin

This class is a child class of User class. This class based on main roll call Admin. It has some important and unique feature to done.

### Methods:

Creat\_user (name, user\_id, password)

- Return type: Void
- Purpose and description: User creation action is execute by this method.

Delete\_user (user\_id)

- Return type: Void
- Purpose and description: This method delete user from server.

Change\_password (user\_id)

- Return type: String
- Purpose and description: Admin can allow to change password or not for any user.

Creat\_role (user\_id, user\_role)

- Return type: void
- Purpose and description: Admin can select the user role ex: who will be the manager?

## Class: Manger

This class is a child class of User class. This class is for manage groups and there works.

### Methods:

Creat\_repository (repo\_name, user\_id, upload\_date)

- Return type: Void
- Purpose and description: This method create repository for a group. And creator only can manage the repository.

Delete\_  sitory (repo\_name)

[Print](#)

- Return type: Void
- Purpose and description: This method can delete repository any time but only creator can delete the repository.

## Class: Groups

Purpose of this class is dealing users group.

### Attributes:

- group (String): This attribute takes the group name.

### Methods:

Creat\_groups (group)

- Return type: Void
- Purpose and description: Group creation action is performed by this method.

Delete\_group (group)

- Return type: Void
- Purpose and description: This method delete group.

Add\_member (user\_id, group)

- Return type: Void
- Purpose and description: Adding member to the group.

remove\_member (user\_id, group)

- Return type: void
- Purpose and description: Remove member from the group.

## Class: Repository

User can create repository and unlimited folder as well as delete folder too. User should be verified by authority and must have permission to access all the folder and file. This user verification and all mentioned functionality is done by this repository class.

### Attributes:

- Repo\_name (String): This string type variable will store the repository name created by user.
- Upload\_date (Date): This variable will contain the upload date of any document or file.
- Folder\_name (String): This string type variable will store the folder name created by user.

### Methods:

Create\_folder (folder\_name)

- Return type: Void
- Purpose and description: Folder creation action is performed by this method.

Delete\_folder (folder\_name)

- Return type: Void

- Purpose and description: Folder deletion action is performed by this method.

[Print](#)

Verify\_user (user\_id)

- Return type: String
- Purpose and description: This method ensure that only verify and permitted user by admin can access the repository and containing folder and also notify admin.

## Class: Message

The message class is where user can communicate with each other by sending text. This class is capable of sending, receiving and deleting text message.

### Attributes:

- message (string): This string type variable will hold the message to carry.

### Methods:

Send\_msg(message)

- Return type: String
- Purpose and description: This method will send message to the desired receiver.

Delete\_msg (message)

- Return type: Void
- Purpose and description: To delete the message this method will be called.

receive\_msg (message)

- Return type: String
- Purpose and description: This method will receive the message from sender.

## Class: Search

Search class will execute search command and go through the database to find out the result.

### Attributes:

- var (String): This string type variable will store the string for user search keyword.

### Methods:

search(var)

- Return type: String
- Purpose and description: This method perform search operation and show results.

## Class: Notification

This class will handle notification system for the whole system by notifying user in different cases.

### Attributes:

- message (String): This message attribute carry the message about notification about what happened in the system for specific action.



**Method:**

Notify\_change (message)

[Print](#)

- Return type: String
- Purpose and description: Notifying admin if any of the user request for password change is performed by this method.

Notify\_doc\_upload (message)

- Return type: String
- Purpose and description: If any kind of document uploaded and previous document updated this method will notify user so that user can also sync update his file.

Notify\_when\_login() (message)

- Return type: String
- Purpose and description: Any kind of unauthorized login attempt is reported to admin by sending notification.

**Class: Backup**

This class will perform data backup and recovery operation both automatically by system and manually by user.

**Methods:**

Backup\_to\_database ()

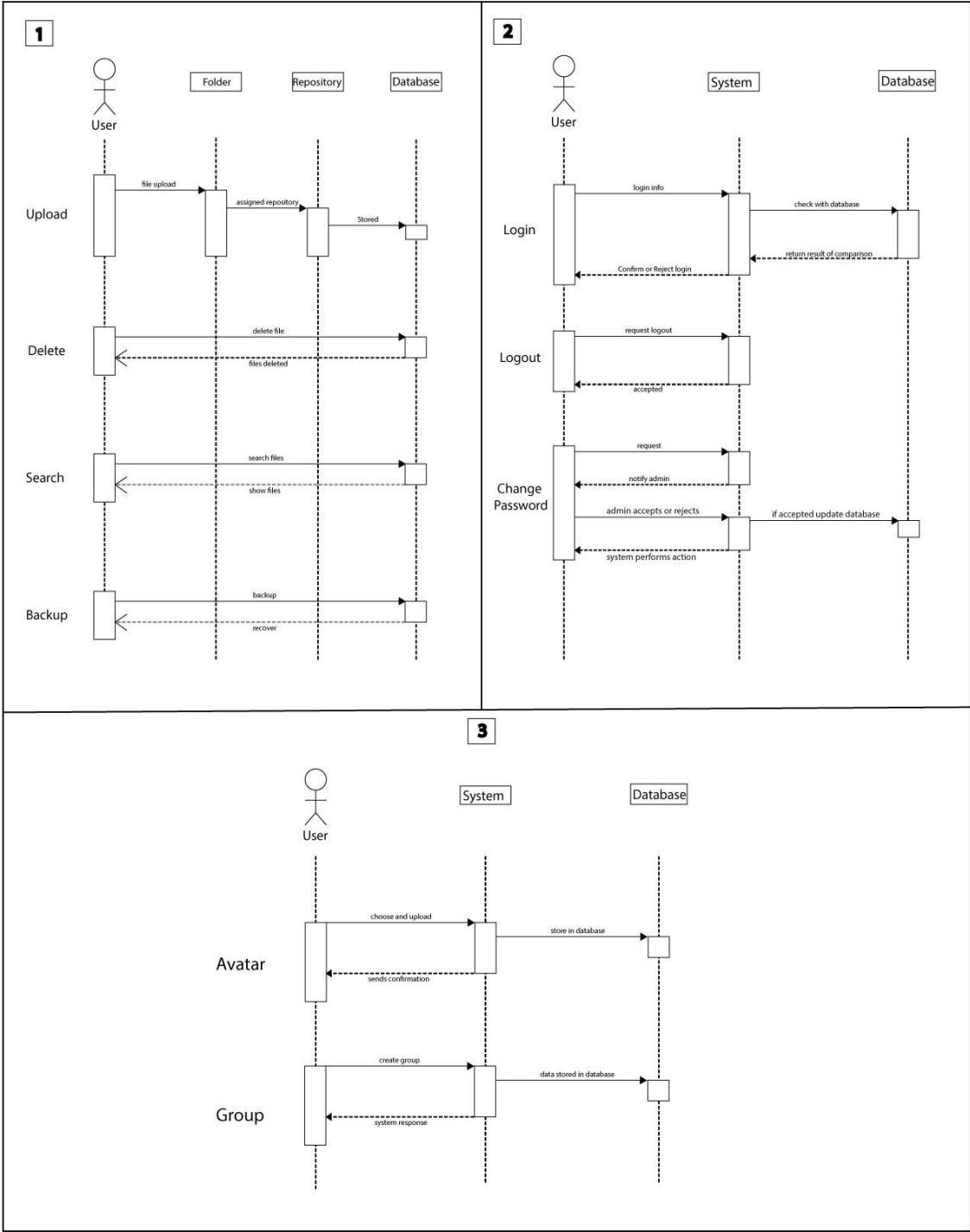
- Return type: void
- Purpose and description: This method will keep the backup of database to a dedicated space by automatically as well as manually by user.

Recover\_from\_db ()

- Return type: Void
- Purpose and description: Backed-up data can be manually recovered from the dedicated space and this method will perform the action.



# Sequence Diagrams



2

User

System

Database

Login

Logout

Change Password

```
sequenceDiagram
    actor User
    participant System
    participant Database

    User->>System: login info
    System->>Database: check with database
    activate Database
    Database->>System: return result of comparison
    deactivate Database
    System->>User: Confirm or Reject login
    deactivate System

    User->>System: request logout
    activate System
    System->>User: accepted
    deactivate System

    User->>System: request
    activate System
    System->>User: notify admin
    deactivate System
    User->>System: admin accepts or rejects
    activate System
    System->>Database: if accepted update database
    activate Database
    Database->>System: 
    deactivate Database
    System->>User: system performs action
    deactivate System
```

3

User

System

Database

Avatar

Group

```
sequenceDiagram
    actor User
    participant System
    participant Database

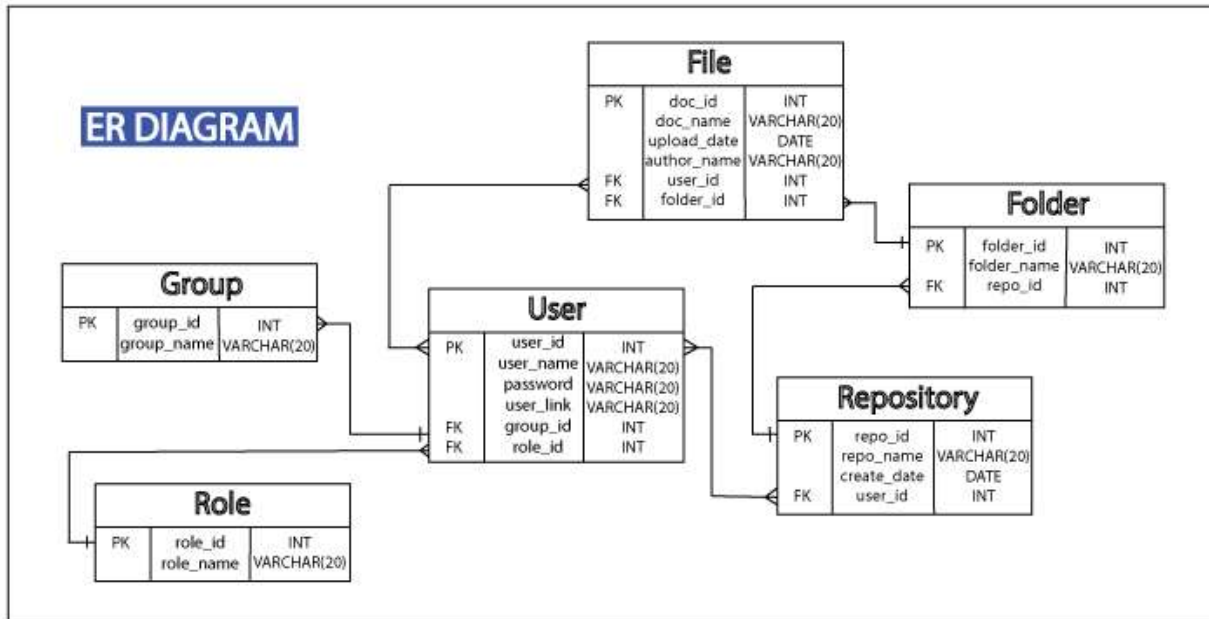
    User->>System: choose and upload
    activate System
    System->>Database: store in database
    activate Database
    Database->>System: 
    deactivate Database
    System->>User: sends confirmation
    deactivate System

    User->>System: create group
    activate System
    System->>Database: data stored in database
    activate Database
    Database->>System: 
    deactivate Database
    System->>User: system response
    deactivate System
```



# Database Design

## ER Diagram



### Purpose of the table:

#### User:

The User table is where the data about the user id, user name, user link, password, group id and role id is stored. Here group ID and role id is foreign keys and user ID is primary key. In this table user have specific id and name, the user can change the password, user also has a group id and have a link to user profile. User have specific role id to clarify the role.

#### Attributes of USER table:

- **user\_id** : Primary Key. This is int type .It holds the id of users.
- **user\_name** : This is string type. It contains the name of user.
- **password**: This is string type . It will store password of the the user.
- **user\_link** : Foreign Key. This is string type. It will store link of the user.
- **group\_id** : This is string type. Specifies the group of the user.
- **role\_id** : Foreign Key. This is string type . It will clarify the role of the user.

#### Repository:

The Repository table is where the data about the repository id, repository name, create date, user id is stored. Here, repository id is primary key and user id is foreign key. In this table, repository have specific id and name , create date of the repository and the user id who create the repository.

#### Attributes of REPOSITORY table:

- repo\_id : Primary Key. This is int type. Every repository have specific id.
- repo\_name : This is string type. Every repository have specific name.
- create\_date: This is date type . It will show the the create date of the repository.
- user\_id : Foreign Key. This is int type. It will show who created the repository and it will verify the specific user of the repository.

Print

## Role:

The Role table is where the information about the role id,role name is stored.It will clarify the position of admin ,employee, manager.

### Attributes of ROLE table:

- role\_id : Primary Key. This is int type. It holds the role id and it is very important because it will specify the role .
- role\_name : Foreign Key. This is string type. It hold the name of role and it clarify the position.

## File:

The file table will store the informations such as document ID,name, upload date, author's name, user id and folder id of files. Here, document id is the primary key. and user id, folder id is foreign key. In this table, Files have unique id, name, the date of upload and the authors name. It also have the user's id who uploaded it and the folder's id in which it belong.

### Attributes of FILE table:

- doc\_id : Primary Key. This is int type .It stores the document's id number.
- doc\_name : This is string type. It contains the name of file.
- upload\_date : This is string type . It store's the date of file upload.
- author\_name : This is string type. It holds the name of the author .
- user\_id : Foreign Key. This is int type. It will store the user id's to show who has access to the file.
- folder\_id: Foreign Key. This is int type. It will store folder id's to show in which folder the file belong.

## Group:

The Group table is where the information about the Group id,role name is stored. Here, group id is the primary key.

### Attributes of GROUP table:

- group\_id : Primary Key. This is int type. It holds the group id and it will specify the group .
- group\_name : This is string type. It hold the name of the group.

## Folder:

The Folder table is where the information about the Folder id, folder name and repository id is stored.It will basically store the folder information. Here, folder id is primary key and repository id is foreign key.

### Attributes of FOLDER table:

- folder\_id : Primary Key. This is int type. It holds the folder id and it will specify the group .
- folder\_name : This is string type. It hold the name of the folder.
- repo\_id : Foreign Key, This is int type. It indicates the repository in which the folder belong.



## Implementation Plan

A developer team of four member are will work simultaneously. As the whole project is done by only four people, so, for designing the SRS (Software Requirements Specification) and SDS (Software Design Specification) whole team works and hold meeting session regularly. After discussing decision will made if any kind of change in necessary. Within first week SRS should be completed and within next 15 days SDS should be completed by holding regular meeting at Google-meet. Shahrukh Fardin is better than everybody in graphics designing. He should design all graphical segment by discussing with rest of the team. Sajib Hossain, Kamrus Samad and Tanvir will do rest of the part by diving work among them with in SRS and SDS submit deadline. Our project will run on multiple platform. We are planning to deploy t his software in two platform Android and Windows. We will develop the software so that it can run on minimum android 4.3 for computer windows seven or later edition that can run HTML5. But for our testing we are using windows 10 and virtual emulator for android. SQLite come with the framework Django. It's easy and free of cost. We will use this as database system and REST API to communicate with the software interface as well as other service. HTML5, CSS, java-script, bootstrap and Django for web-based application and JAVA for android development. Designing user interface, database and UML diagram whole team works by holding meeting every-day. After completing design Kamrus Samad and Shahrukh Fardin will work on web-based application. On the other hand, Sajib Hossain and Tanvir will develop android application using Android studio within next 25 days (can be change with deadline). As same user interface will be deployed in multiple platform, so, everybody will follow same user interface design and same method name for well coordination.