DigiLocker

1.0. Introduction

1.1. Purpose

In today's digital era there is a need of documents and data in form of soft copies, as it is easy to transfer and reuse them whenever required. Managing such important data is essential so that it's easy for the people to retrieve them.
The Digi Locker is a web-based application that helps you to store and manage your documents in an organized manner.

Your data is stored using the latest encryption technology making it highly secure and can be accessed by you anytime through the application.

This document is meant to delineate the features of DigiLocker, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2. Scope of Project

Initial functional requirements will be: -

- Secure registration and profile management facilities for User.
- E-mail facility for activation of User Account.
- Creating a Portal where User can upload 'n' no. of files and can download them whenever required.
- Browsing through the uploaded files of different category.
- User can view and update personal profile whenever required.
- DigiLocker helps to manage User's files in a logical manner and make them secure by using strong security system such as Encryption technology.

Initial non-functional requirements will be: -
Secure access of confidential data (user's data).

24 X 7 availability.

Better component design to get better performance at peak time

In addition to the above-mentioned points, due to the highly evolving nature of the project, the following are planned to be delivered if deemed necessary:

Easy to access and use the website.

Term	Definition
Database	Collection of all the information monitored by this system.
Field	A cell within a form.
Member	A member of the website listed in the User database.
User	Anyone visiting the site to interact with the web application.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

1.3. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

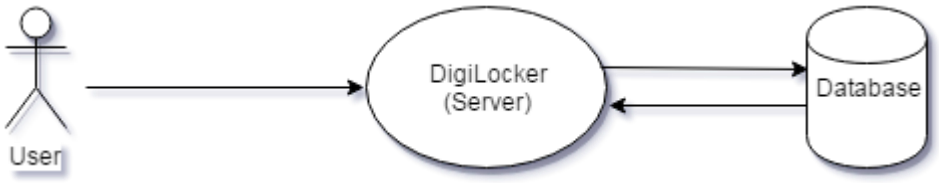
Both sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different language.

2.0. Overall Description

2.1 System Environment

Figure 1 - System Environment

The DigiLocker has only one active actor and one cooperating system on the server that internally interacts with the database.



2.2 Functional Requirements Specification

This section outlines the use cases for each of the active Customers separately. The User, and the reviewer have only one-use case apiece while the admin is main actor in this system.

Use case: **Registration**
Diagram:

Brief Description
The user should be able to register successfully after filling the registration form with valid data and clicking on register button.

Initial Step-By-Step Description
Before this use case can be initiated, the User has already accessed the DigiLocker Website.

1. The User clicks on the Register button.
2. The system displays the Registration form.
3. The User enters all required credentials in the specified text boxes.
4. The User chooses the country from the given dropdown list and clicks on register button.
5. The system sends an account activation link to the email entered by the user.
6. The User clicks on the link sent to his email ID.
7. The system activates the user account.

Use case: **Login**
Diagram:

Brief Description
The User should be able to login to his registered account with valid user credentials on the DigiLocker website.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be registered with the DigiLocker Website.

1. The User accesses the home page of the website and clicks on the login button.
2. The System opens the login form.
3. The User enters the registered username and password in the textboxes provides for the same and click on the Login button.
4. The System verifies the user credentials and gives user access to the account.

Use case: **Upload**
Diagram:

Brief Description
The User should be able to upload a file with a registered account.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User clicks on the upload button.
2. The System opens the upload form.
3. The User chooses the file to be uploaded through the form, chooses the category to be uploaded, fill all required credentials and click on upload.
4. The System uploads the file.

Use case: **Download**
Diagram:

Brief Description
The User should be able to download a file by clicking on the download button.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User chooses the category from the view files option.
2. The System opens the chosen category.
3. The User clicks on the download button of the file to be downloaded.
4. The System downloads the file on the local system of the user.

Use case: **View Profile Details**
Diagram:

Brief Description
The User should be able to view his profile details by clicking on the profile button.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User clicks on the profile button that has the name of the user.
2. The System opens the user profile details.

Use case: **Edit Profile Details**
Diagram:

Brief Description
The User should be able to edit his profile details.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User clicks on the edit button on the profile details page or choose the edit Profile option from the dropdown arrow box.
2. The System opens the editable user profile.
3. The user edits the textfield(s) and click on Update button.

Use case: **Change Password**
Diagram:

Brief Description
The User should be able to change his registered account password.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User chooses the settings option from the dropdown arrow menu box.
2. The System opens the settings option.
3. The User clicks on the change password option.
4. The System opens the change password form.
5. The user enters the old password and the new password in the textbox provided and clicks on OK.
6. The System changes the user account password with a success message.

Use case: **Delete a File.**
Diagram:

Brief Description
The User should be able to delete an uploaded file.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User chooses the category from the view files option.
2. The System opens the chosen category.
3. The User clicks on the delete button of the file to be deleted.
4. The System moves the file into Trash.
5. The User may click on Trash option from the category list and click on the delete button of the file.
6. The System deletes the file permanently.

Use case: **Restore a deleted File.**
Diagram:

Brief Description
The User should be able to delete an uploaded file.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User chooses the category from the view files option.
2. The System opens the chosen category.
3. The User click on Trash option from the category list and click on the undo button of the file.
4. The System restores the file.

Use case: **Deactivate User Account**
Diagram:

Brief Description
The User should be able to deactivate his active account.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User chooses the settings option from the dropdown arrow menu box.
2. The System opens the Settings page.
3. The user clicks on Deactivate My Account.
4. The System show an alert box with appropriate conformation message.
5. The User chooses "Yes".
6. The System deactivates the user account and opens the home page with success message.

Use case: **Logout**
Diagram:

Brief Description
The User should be able to deactivate his active account.

Initial Step-By-Step Description
Before this use case can be initiated, the User must be logged in with a registered account on DigiLocker Website.

1. The User Clicks on the dropdown arrow menu button.
2. The System opens the menu.
3. The User chooses the Logout option.
4. The System Logs out the user and opens the home page with appropriate success message.

2.3 User Characteristics

The User is expected to be Internet literate and be able to use the website. The main screen of the DigiLocker Website will have a link to Login and Registration Page.
The detailed look of these pages is discussed in section 3.2 below.

2.4 Non-Functional Requirements

The DigiLocker will be on a server with high speed Internet capability. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the User's connection will depend on the hardware used rather than characteristics of this system.

3.0. Requirements Specification

3.1 External Interface Requirements

The only link to an external system is the link to the DigiLocker Database to register and verify the membership of a User.

All the functionality of the website uses the database to store and retrieve User's data.

For example, The *Login* use case sends the User ID to the Database and a Boolean is returned denoting membership status. The *Upload* use case sends a requests to upload the User's file in database. Similarly, the Download use case sends a request to the database to download a specific file the User wants to download.

3.2 Detailed Non-Functional Requirements

3.2.1 Logical Structure of the Data

The logical structure of the data to be stored in the database is given below.

3.2.2 Hardware Requirement:

- PROCESSOR : Pentium IV & Above.
- RAM : 1 GB & Above.
- HARD DISC SPACE: 40 GB & above.
- MONITOR : Color

3.2.3 SOFTWARE REQUIRMENT:

OPERATING SYSTEM : Windows Vista/7 and all above (but Must be Microsoft OS)

IDE : MyEclipse Enterprise Workbench 8.6

Web Server : Apache Tomcat
Frameworks : Servlet API, Java Server Pages (JSP)
Technology : Java EE
Language : Java
Database : Oracle 11g

3.3 Detailed-Functional Requirement:

3.3.1. Users will be able to create accounts to store their user profiles, configure contact information,

3.3.2 Upload their personal files and documents, view their uploaded files according to their category, and download them whenever required. User will be able to register, log in, and log out of their accounts. Furthermore, User can delete the uploaded files which will be moved to the Trash by the system.

3.3.3 The User can undo delete those files from the Trash or delete them permanently from the Trash.

3.3.4 Uploaded files will be stored in different categories. The upload mechanism will allow User to add new files in the categories with the specific file details.

3.3.5 Security

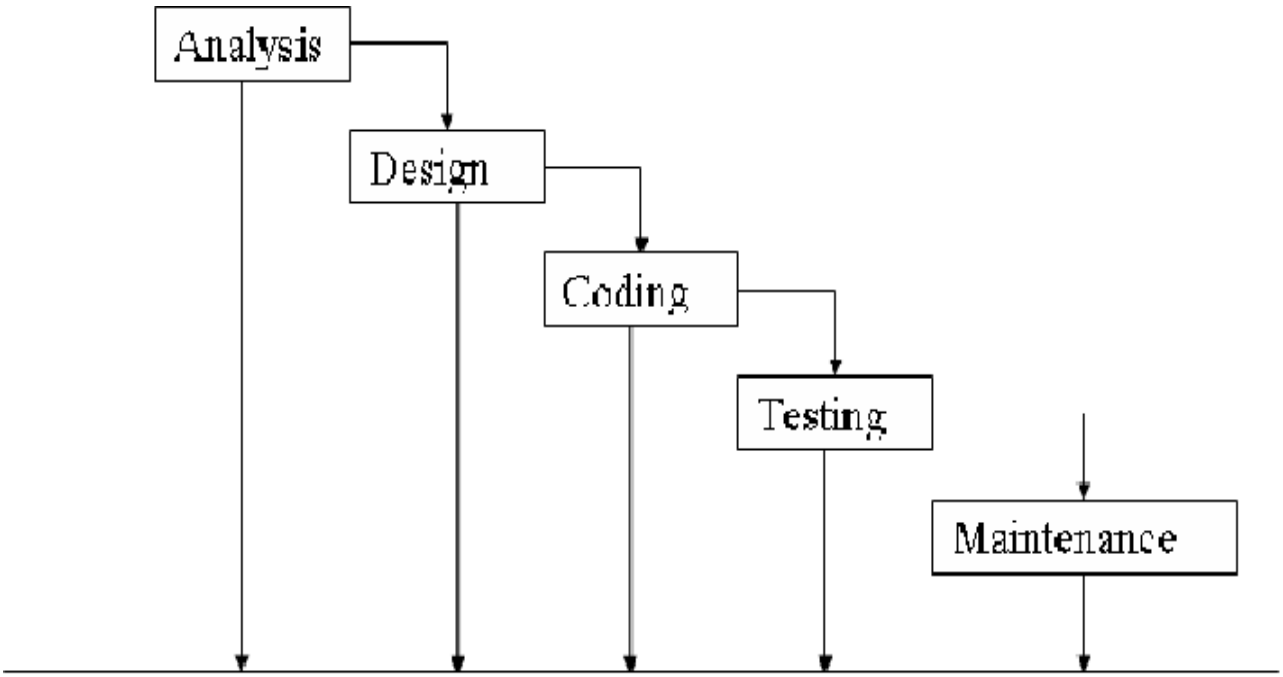
The server on which the DigiLocker resides will have its own security to prevent unauthorized *write*/*delete* access. There are restriction on *read access*. The use of email by the System is on the server and thus is internal to the system.

The PC on which the User resides will have its own security. There are special protection built into this system such as user validation and verification before accessing the system.

The system also has feature like encryption and decryption to secure the important user data.

Software Life Cycle Model

In order to make this Project we are going to use Classic LIFE CYCLE MODEL .Classic life cycle model is also known as WATER FALL MODEL. The life cycle model demands a Systematic sequential approach to software development that begins at the system level and progress through analysis design coding, testing and maintenance.



The Classic Life Cycle Model

The waterfall model is sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception initiation, Analysis, Design (validation), construction. Testing and maintained.

1) System Engineering and Analysis:-

Because software is always a part of larger system work. Begins by establishing requirement for all system elements and Then allocating some subset of these requirement to the software system Engineering and analysis encompasses the requirement gathering at the system level with a small amount of top level design and analysis.

2) Software requirement Analysis: =

The requirement gathering process is intensified and focused specifically on the software. Requirement for the both system and software are discussed and reviewed with the customer. The customer specifies the entire requirement or the software and the function to be performed by the software.

3) Design:

Software design is actually a multi-step process that focuses on distinct attributes of the program data structure, software architecture, procedural detail and interface of the software that can be assessed or quality before coding begins. Like requirement the design is documented and becomes part of the software.

4) Coding:

The design must be translated into a machine-readable form. The coding step performs this task. If design is programmed in a detailed manner, coding can be accomplished mechanically.

5) Testing:

Once code has been generated programmed testing begins. The testing process focuses on the internals of the software ensuring that all statement have been tested and on the functional externals hat is conducting tests to uncover the errors and ensure that defined input will produce the results that agree with the required results.

Unit testing: -

In computer programming, Unit testing is software Verification and validation method where the programmer gains confidence that individual units of source code are fit to use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual programmed, function, procedure, etc. while in object-oriented programming, the smallest Unit is a class, which may belong to a base/super class abstract class or derived/child class.

Benefits:

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict written contract that the piece of code must satisfy.

Documentation: -

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the tests to gain a basic understanding of the unit API.

Limitation of unit testing:

Testing cannot be expected to catch error in the program –It is impossible to evaluate all execution paths for all but the most trivial programs. The same is true for unit testing. Additionally, by unit testing only types the functionality if the units themselves.

6) Maintenance:

Software will undoubtedly undergo change after it is Delivered to the customer. Change will occur because errors have been encountered because the software must be able adopted to accommodate changes in its external environment because the customer requires functional or performance enhancement enhancements. The classic life cycle is the oldest and most widely used paradigm or software engineering

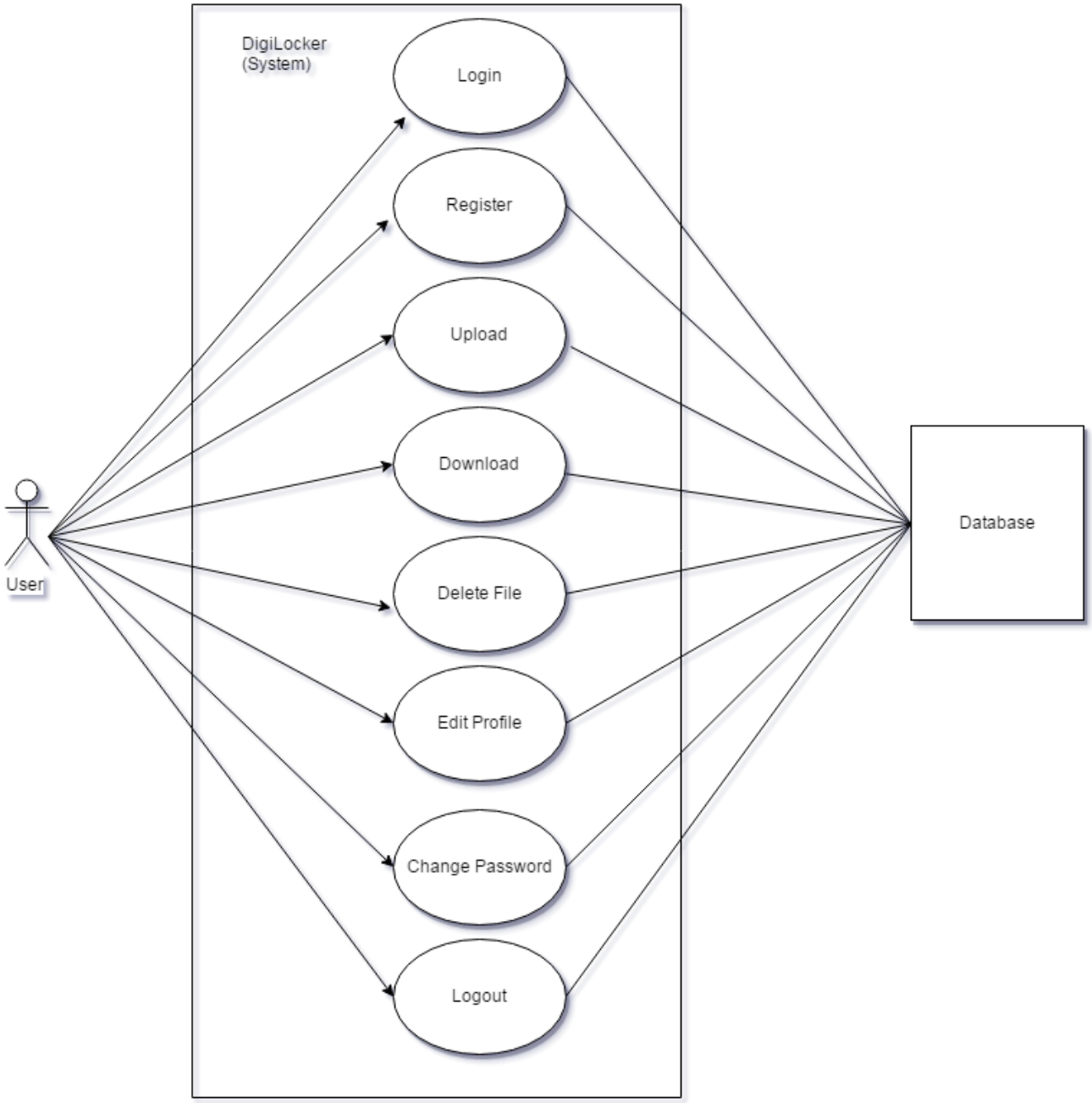
Conclusion

This project “DigiLocker”, helps to manage important documents in an organized manner. Now, it can be done using this software and hence, it makes it easy for the User to store his/her data into a Logical Structure that can be stored in the DigiLocker database on the server; also making critical and confidential data secure. This data can be accessed easily from any web browser anytime by the user.
By working on this project, I learned many new skills that are very important in the development of a software.

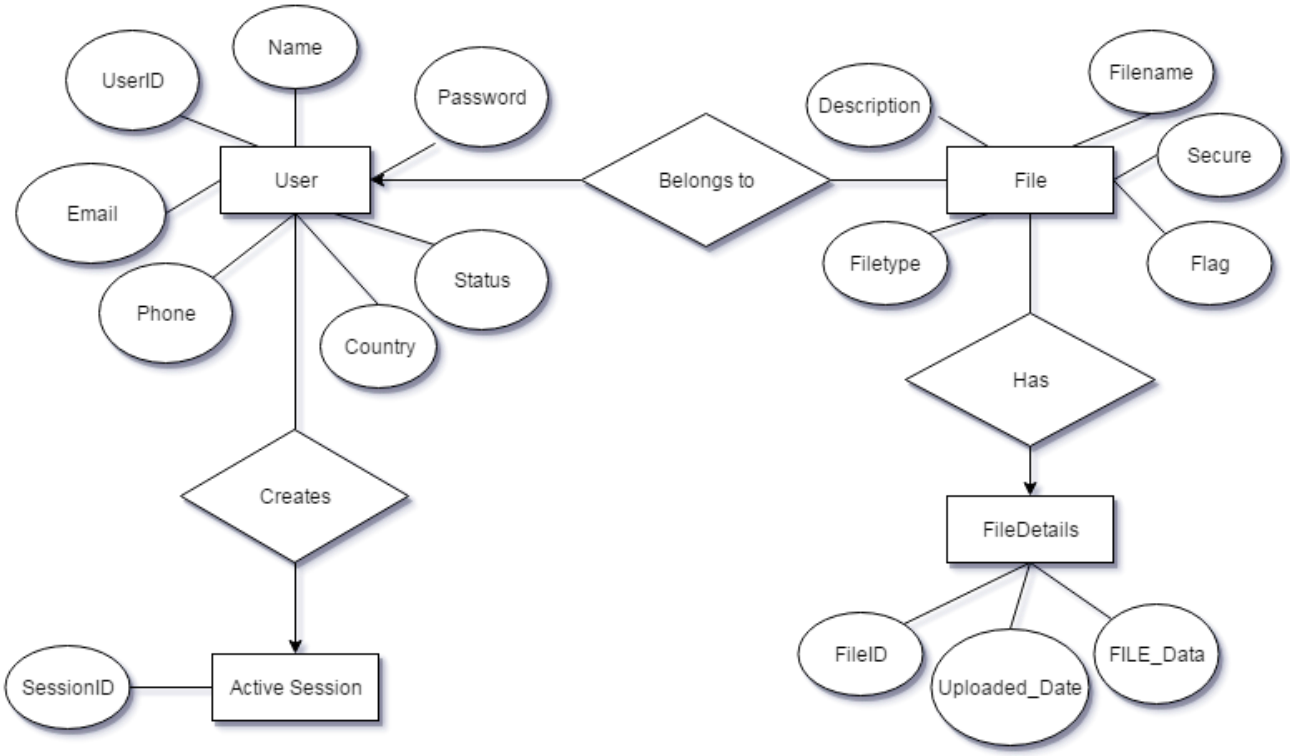
Bibliography

- MK Yonk Url: http://www.mkyong.com/
- W3Schools Url: http://www.w3schools.com/
- StackOverflow Url: http://stackoverflow.com/
- Google Url: http://www.google.com/

Use Case: -

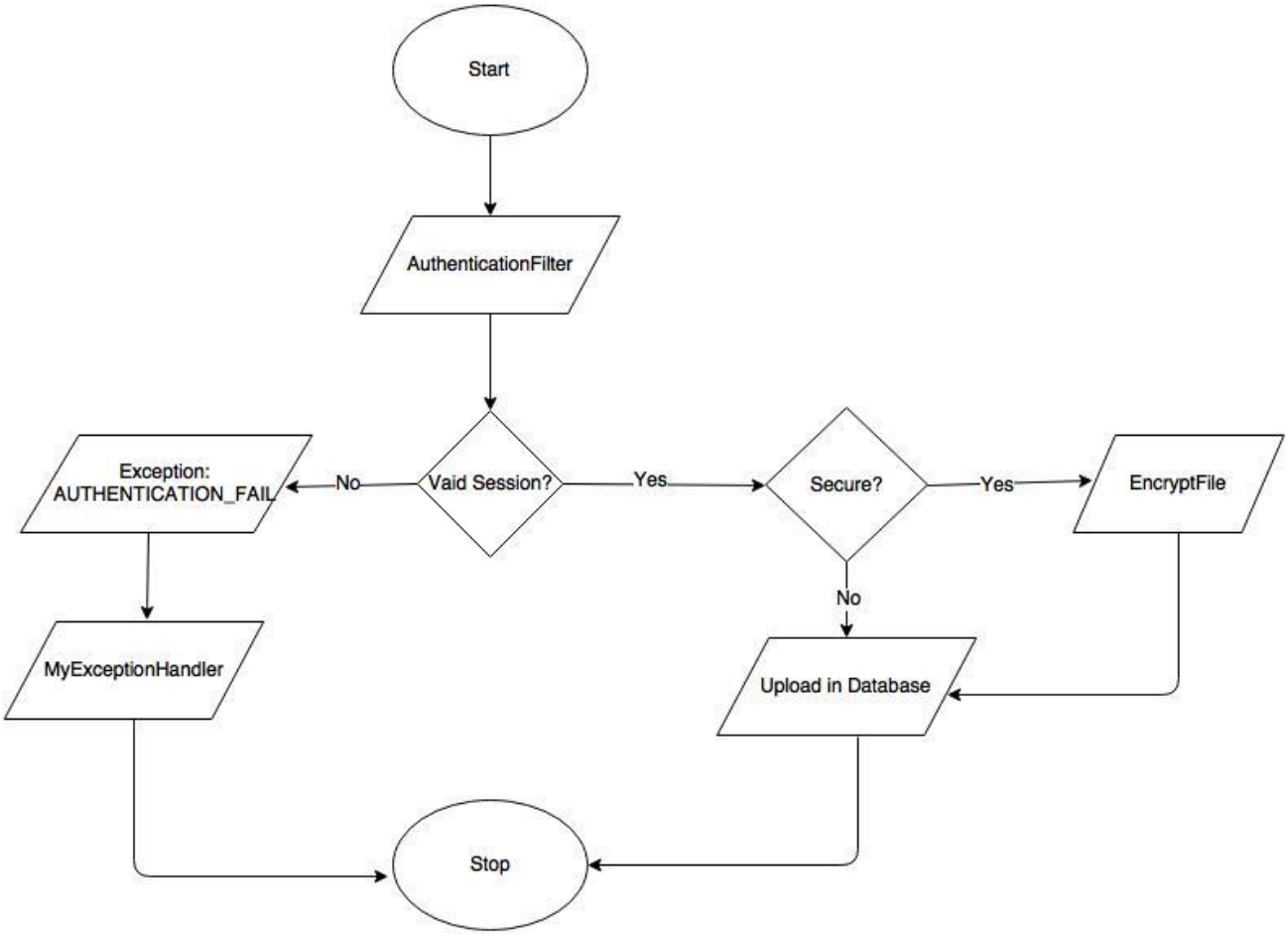

DigiLocker
(System)
Login
Register
Upload
Download
Delete File
Edit Profile
Change Password
Logout
User
Database

ER Diagram:

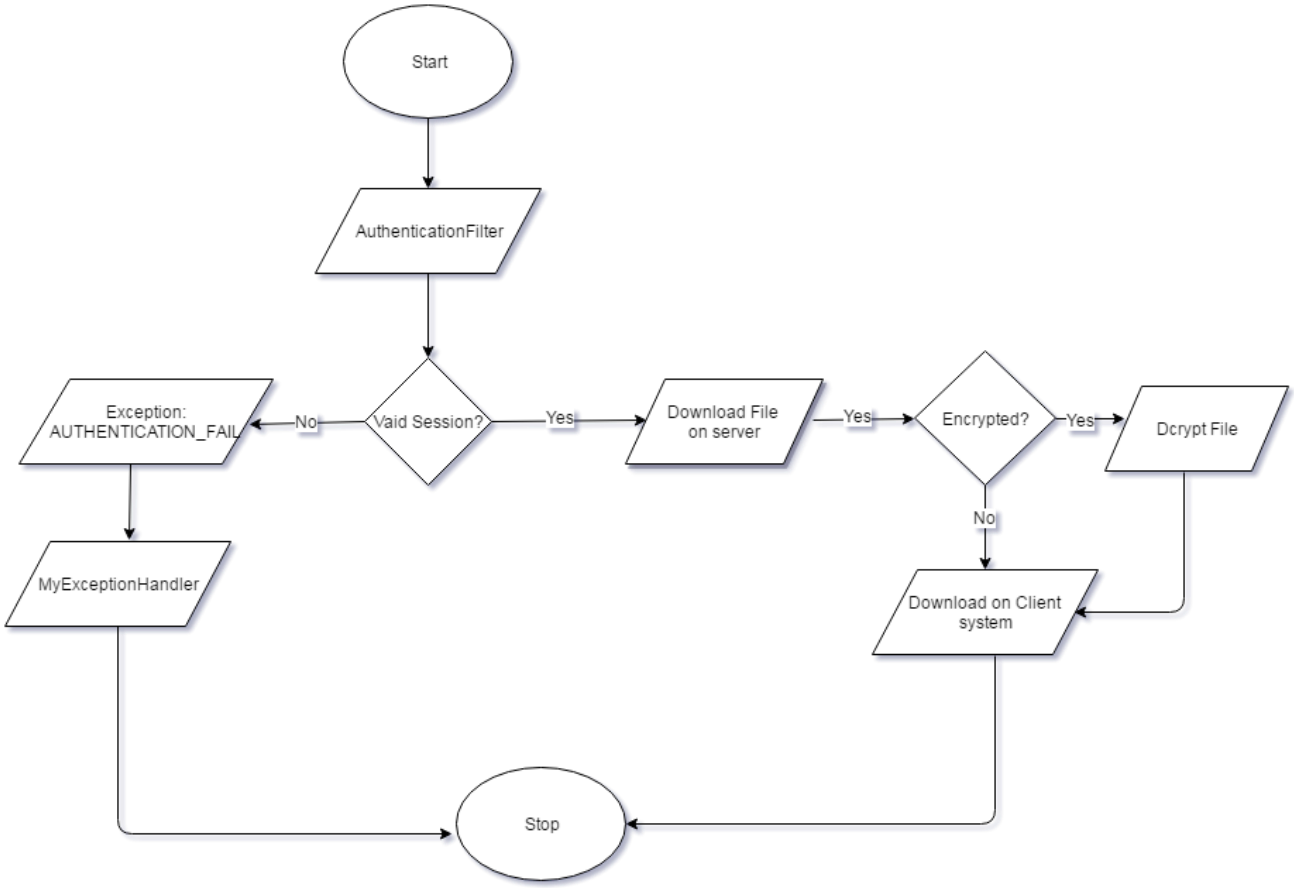

Name
UserID
Password
User
Email
Phone
Status
Country
Creates
SessionID
Active Session
Belongs to
Description
Filename
Secure
File
Filetype
Flag
Has
FileDetails
FileID
Uploaded_Date
FILE_Data

Flow Diagram:

1) Uploading File



2) Downloading File



Database Tables

1) ACTIVE_SESSION

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	USERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	SESSIONID	VARCHAR2(40 BYTE)	No	(null)	2	(null)

2) CATEGORY

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CATEGORY_ID	NUMBER	No	(null)	1	(null)
2	CATEGORY_NAME	VARCHAR2(40 BYTE)	No	(null)	2	(null)

3) FILE_DETAILS

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FILE_ID	VARCHAR2(50 BYTE)	No	(null)	1	(null)
2	UPLOAD_DATE	DATE	No	(null)	2	(null)
3	FILE_DATA	BLOB	No	(null)	3	(null)

4) FILE_INFO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	USERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	FILENAME	VARCHAR2(40 BYTE)	No	(null)	2	(null)
3	CATEGORY_ID	NUMBER	No	(null)	3	(null)
4	DESCRIPTION	VARCHAR2(40 BYTE)	Yes	(null)	4	(null)
5	FILETYPE	VARCHAR2(20 BYTE)	No	(null)	5	(null)
6	FILE_ID	VARCHAR2(50 BYTE)	No	(null)	6	(null)
7	SECURE	VARCHAR2(20 BYTE)	No	(null)	7	(null)
8	FLAG	NUMBER	No	1	8	(null)

5) USER_INFO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	USERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	EMAIL	VARCHAR2(40 BYTE)	No	(null)	3	(null)
4	PHONE	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	NAME	VARCHAR2(20 BYTE)	No	(null)	5	(null)
6	COUNTRY	VARCHAR2(20 BYTE)	No	(null)	6	(null)
7	STATUS	VARCHAR2(10 BYTE)	No	(null)	7	(null)

6) TOKEN

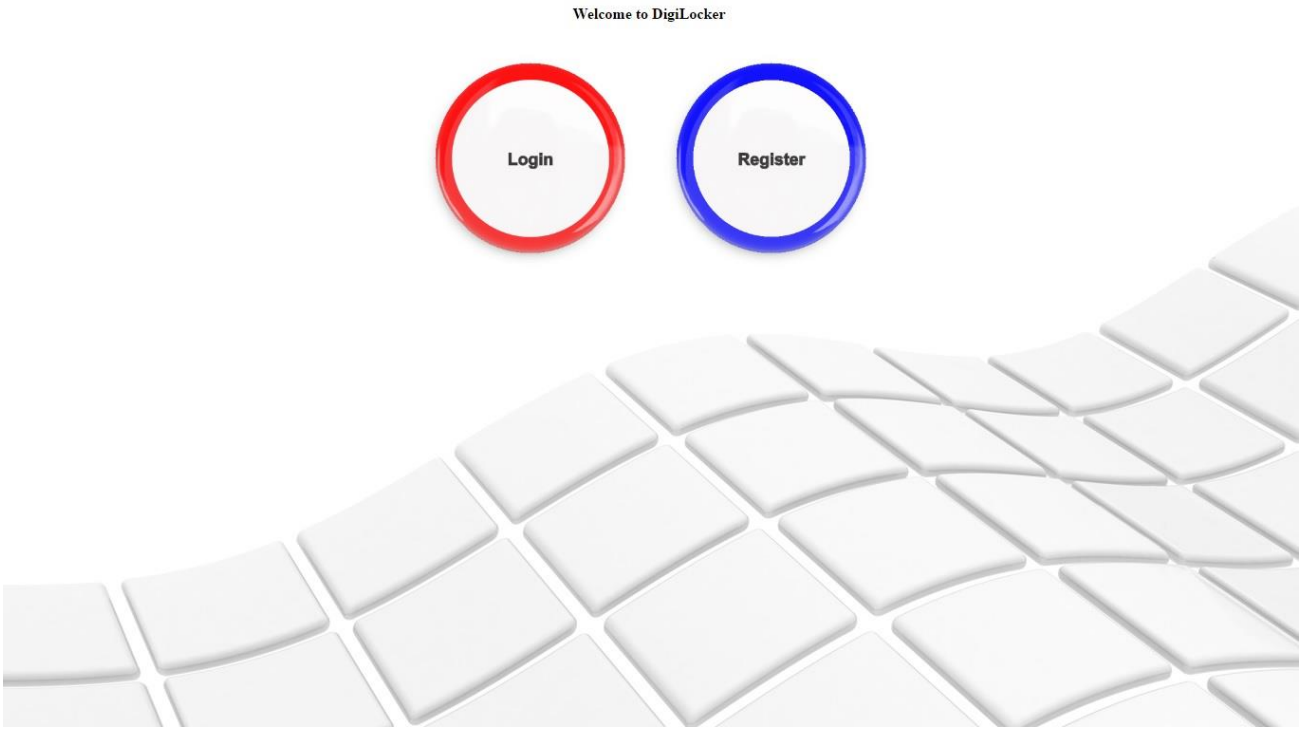
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TOKENID	NUMBER(20,0)	No	(null)	1	(null)
2	USERID	VARCHAR2(20 BYTE)	No	(null)	2	(null)

7) SESSION_HISTORY

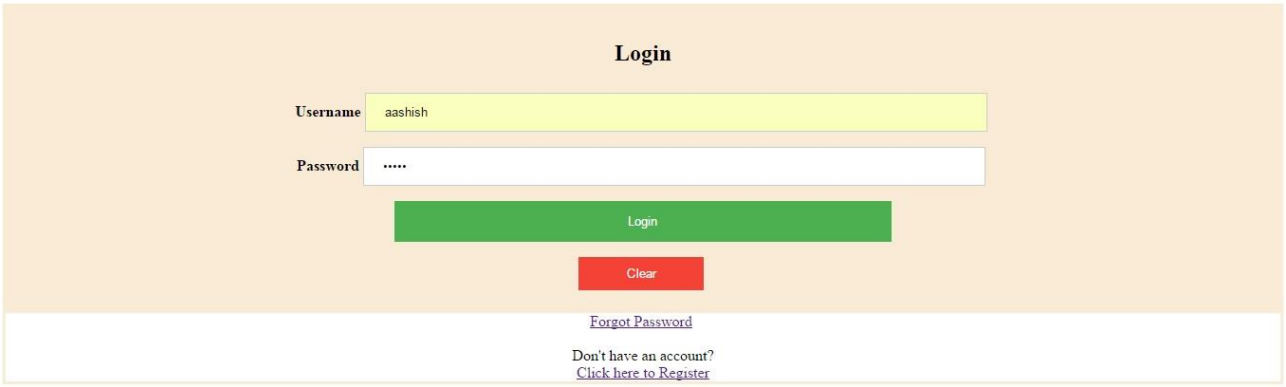
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	USERID	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	SESSIONID	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	LOGIN_TIME	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	LOGOUT_TIME	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)

DigiLocker Screenshots

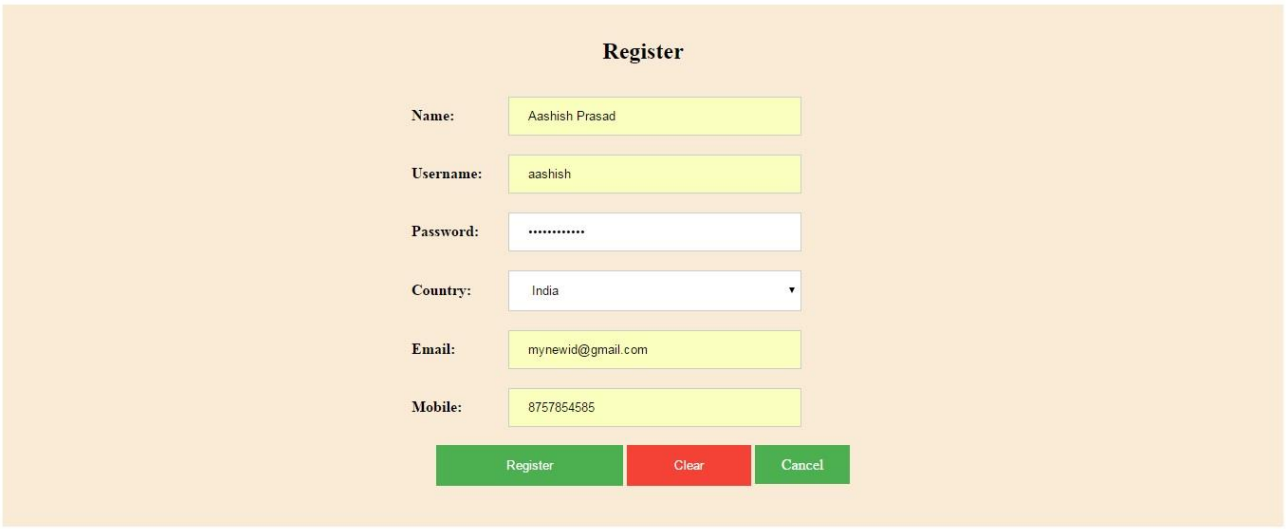
1) Home Page



2) Login Page

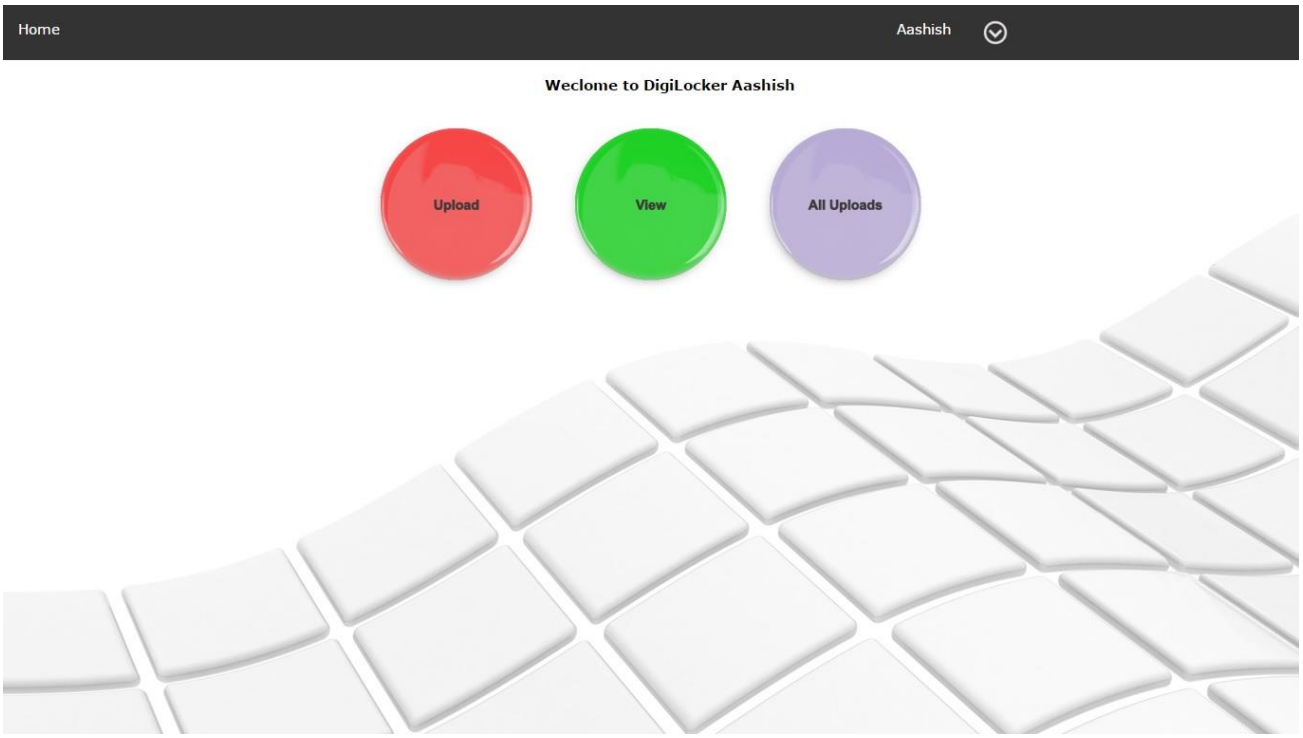


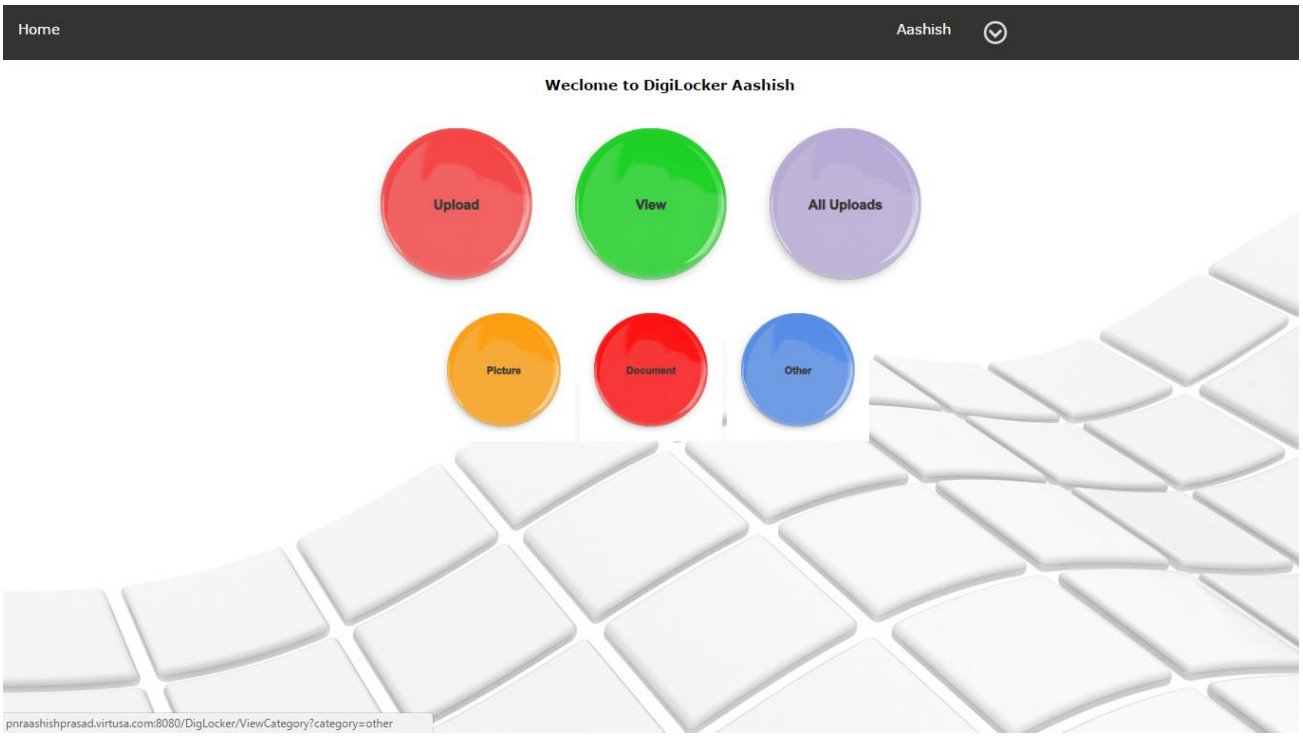
3) Registration Page

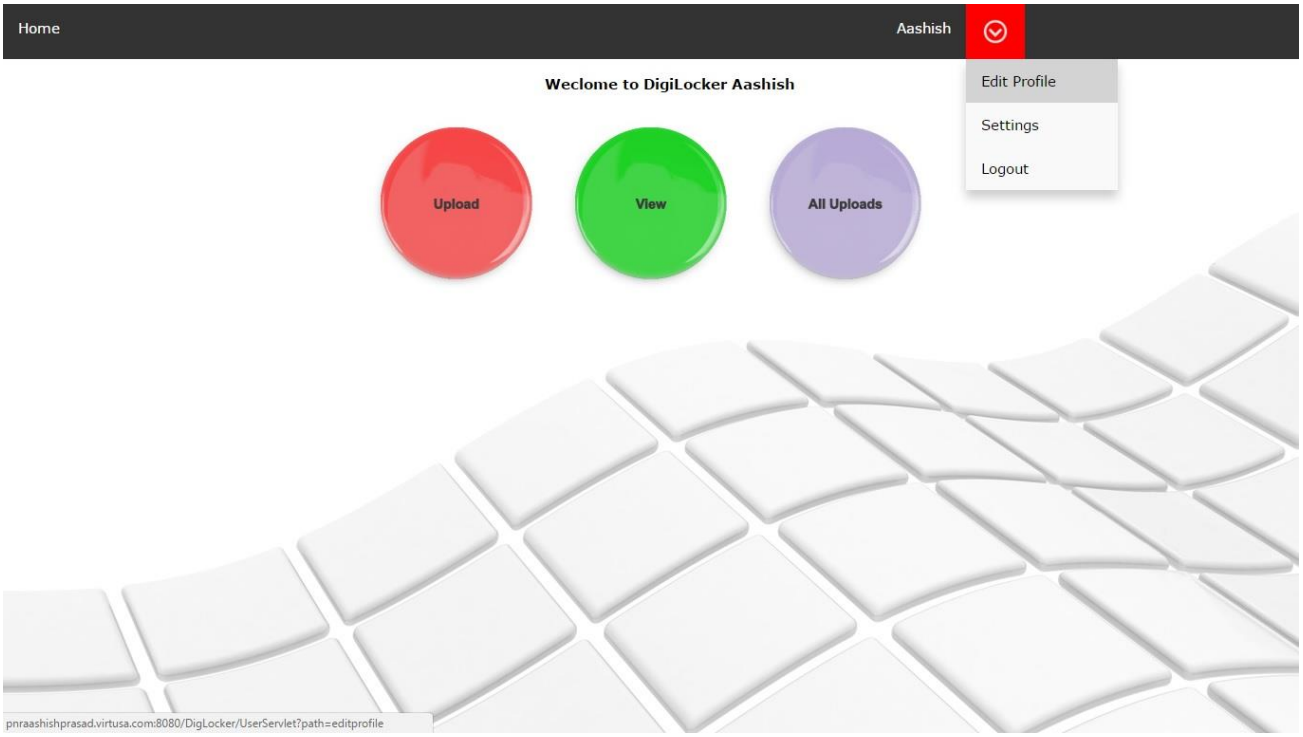


4) Forget Password

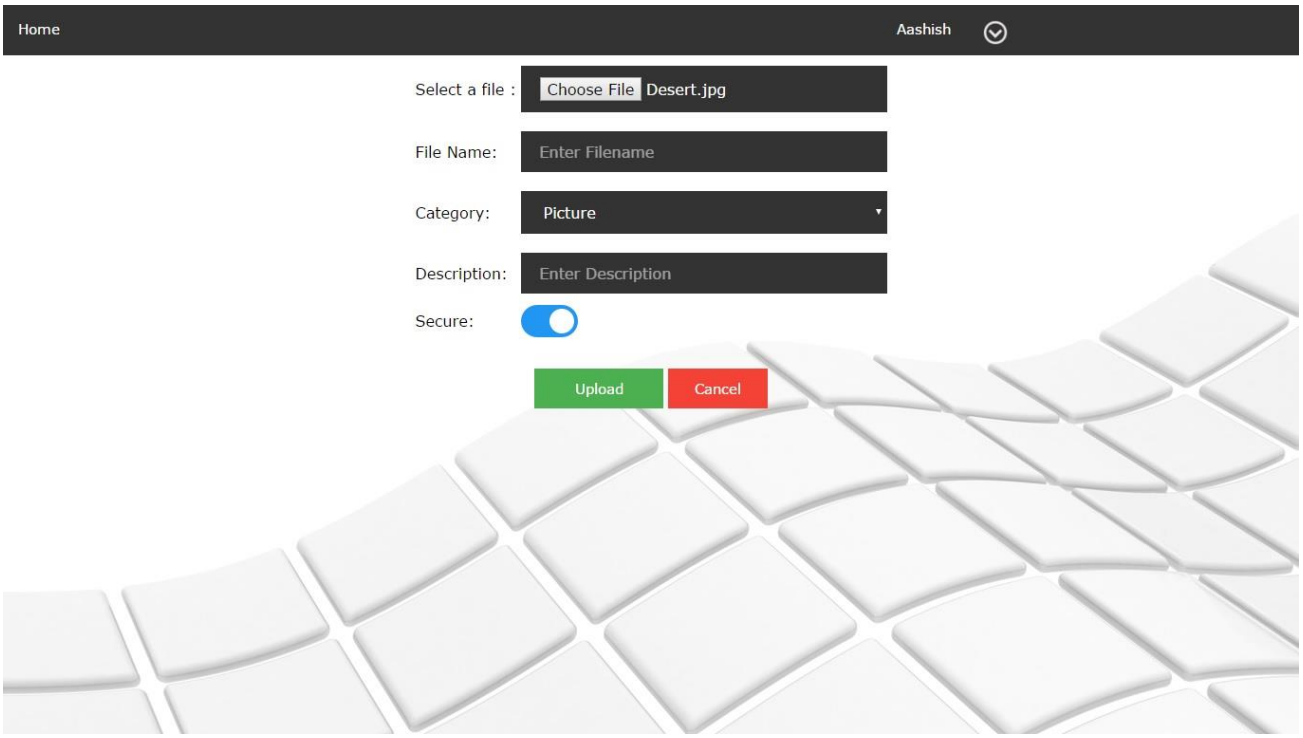
5) Welcome Page



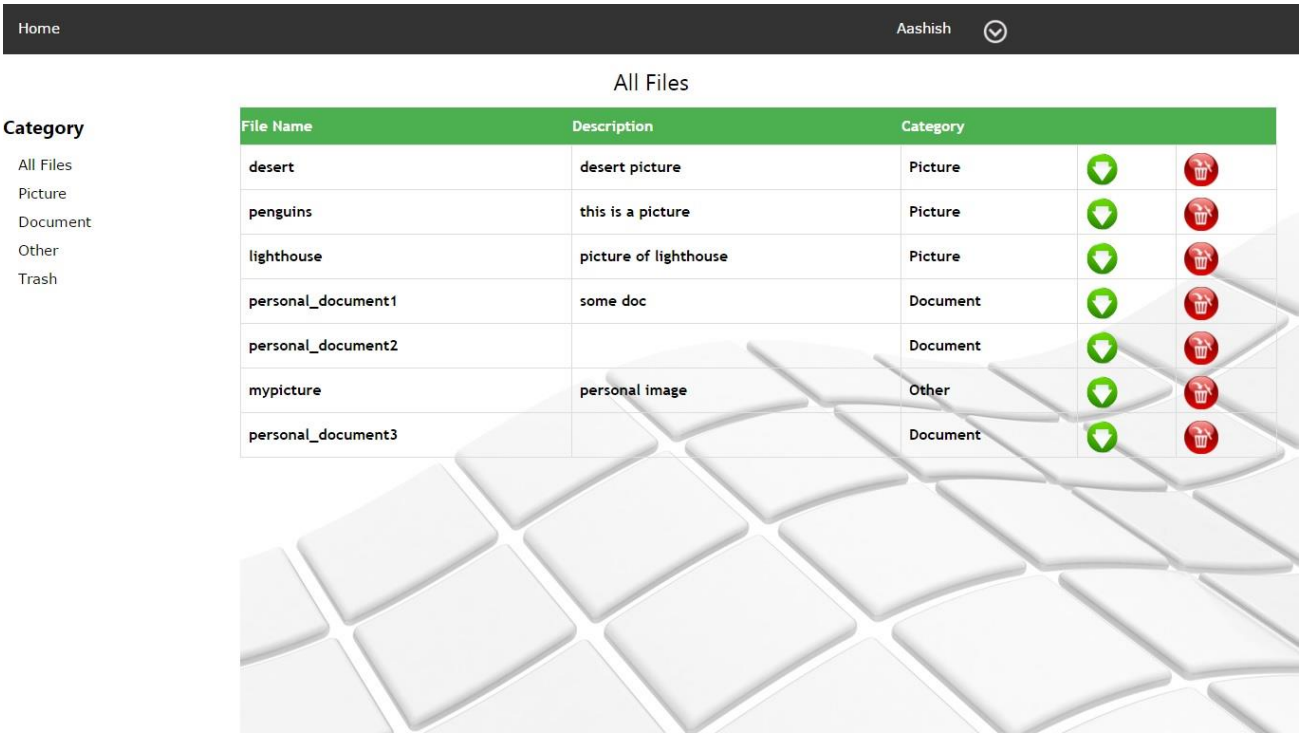

Home
Aashish
Weclome to DigiLocker Aashish
Upload
View
All Uploads
Picture
Document
Other
pnraashishprasad.virtusa.com:8080/DigLocker/ViewCategory?category=other

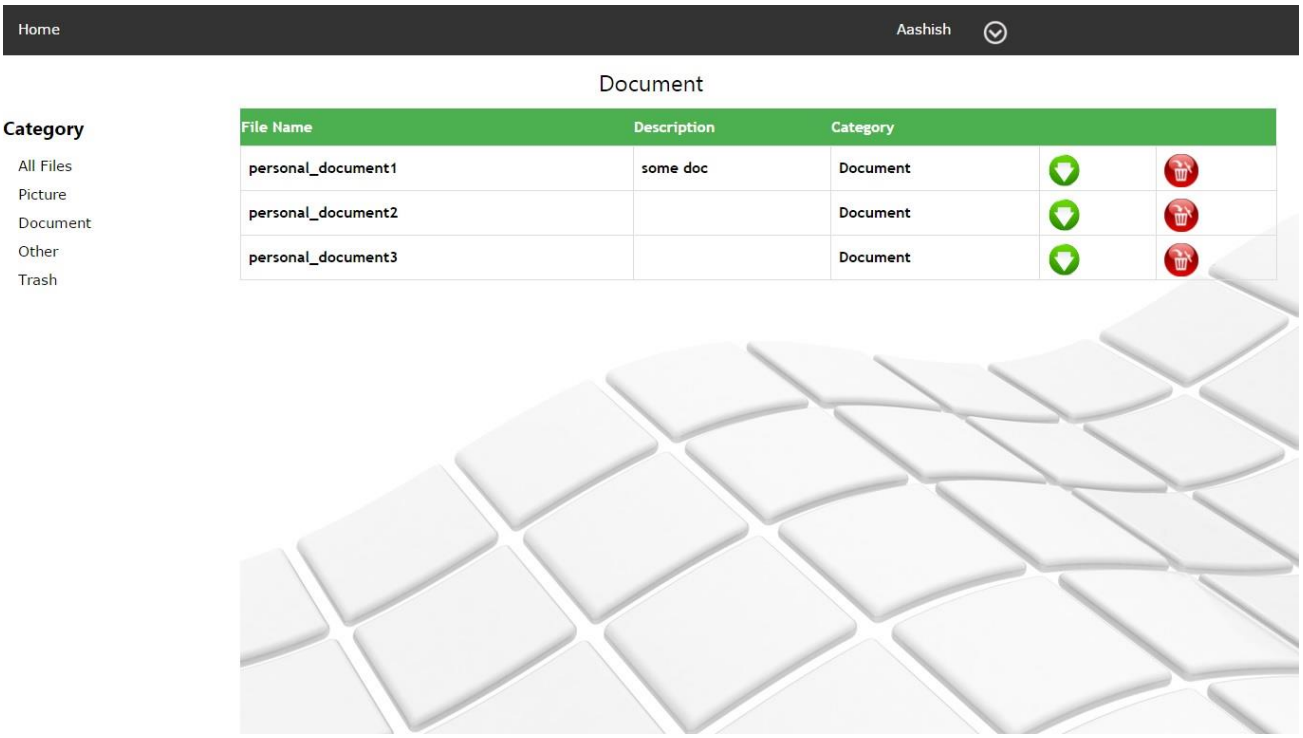

Home
Aashish
Edit Profile
Settings
Logout
Weclome to DigiLocker Aashish
Upload
View
All Uploads
pnraashishprasad.virtusa.com:8080/DigLocker/UserServlet?path=editprofile

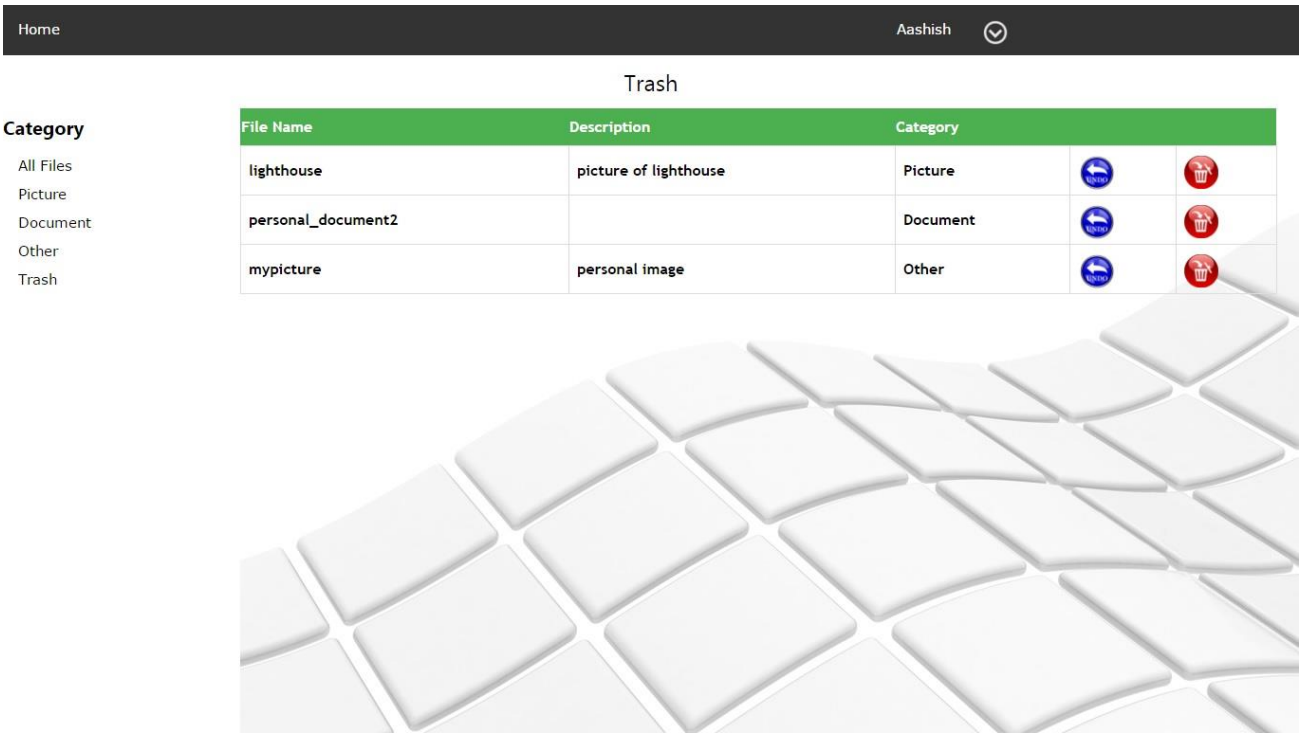
6) Upload File



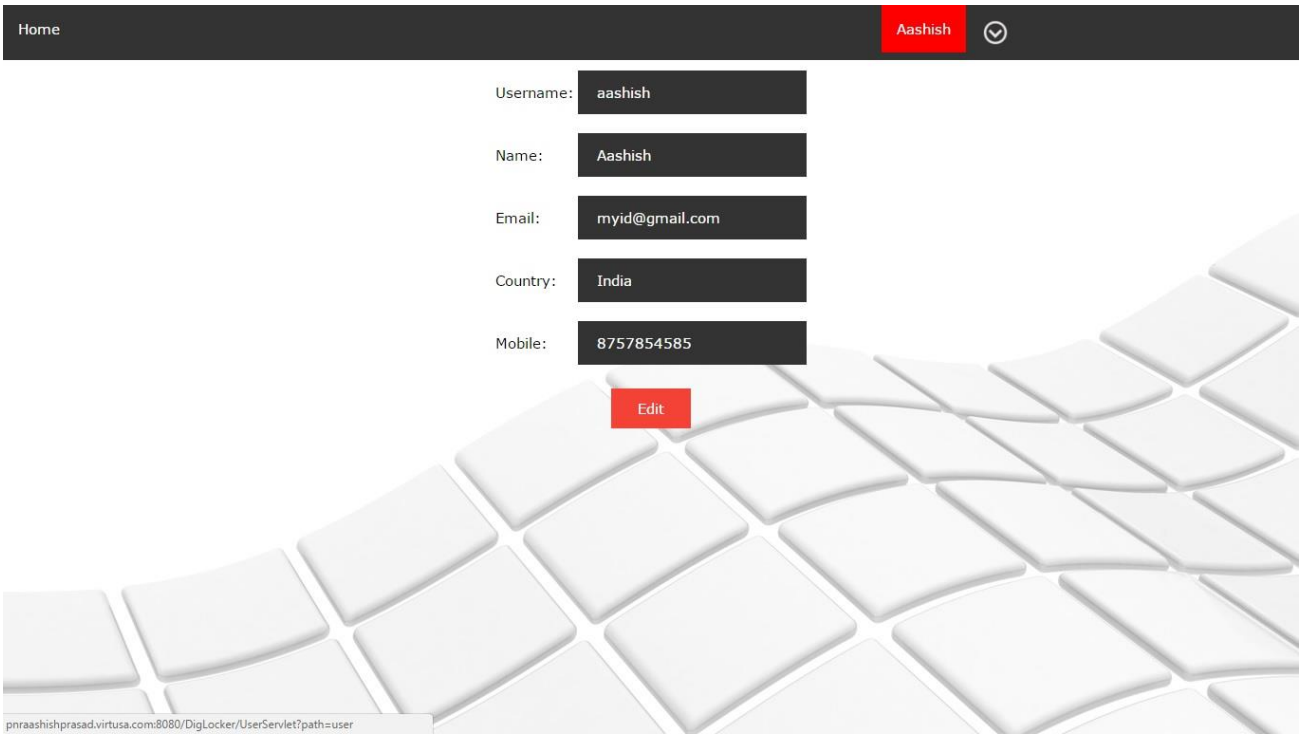
7) Category View



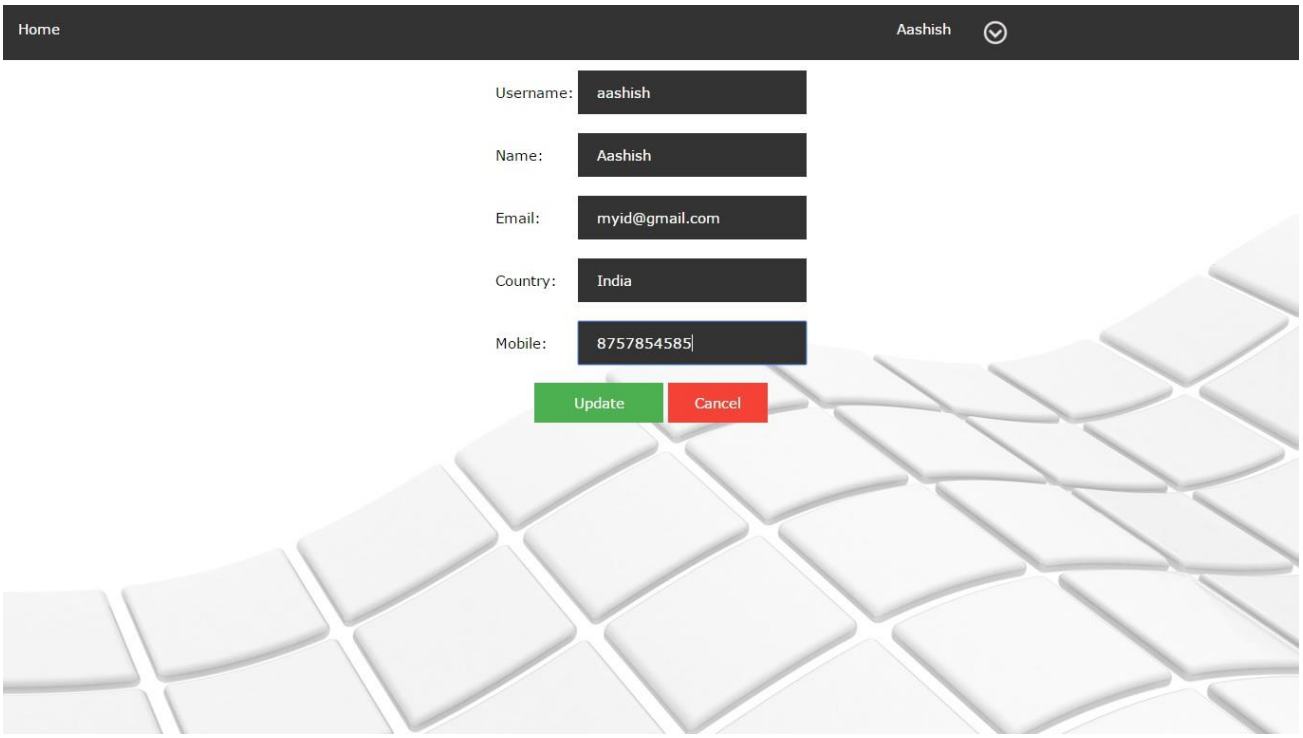




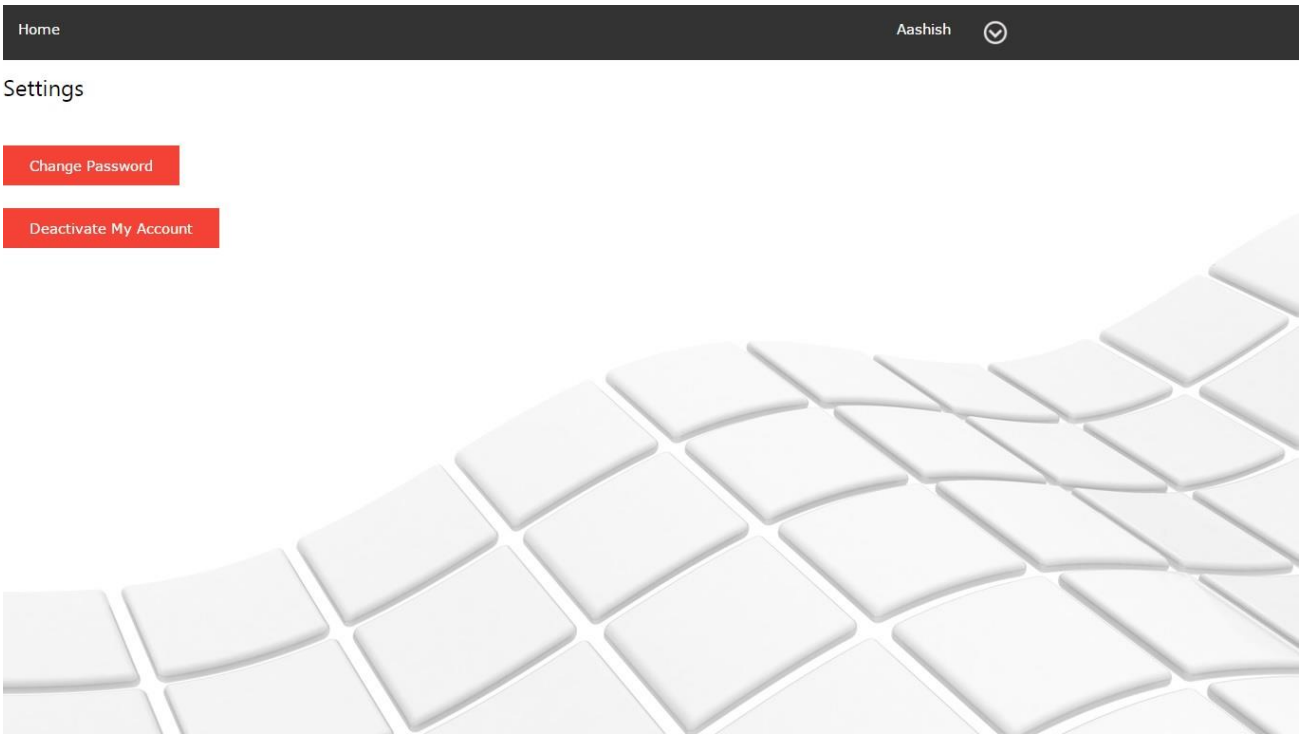
8) Profile Details



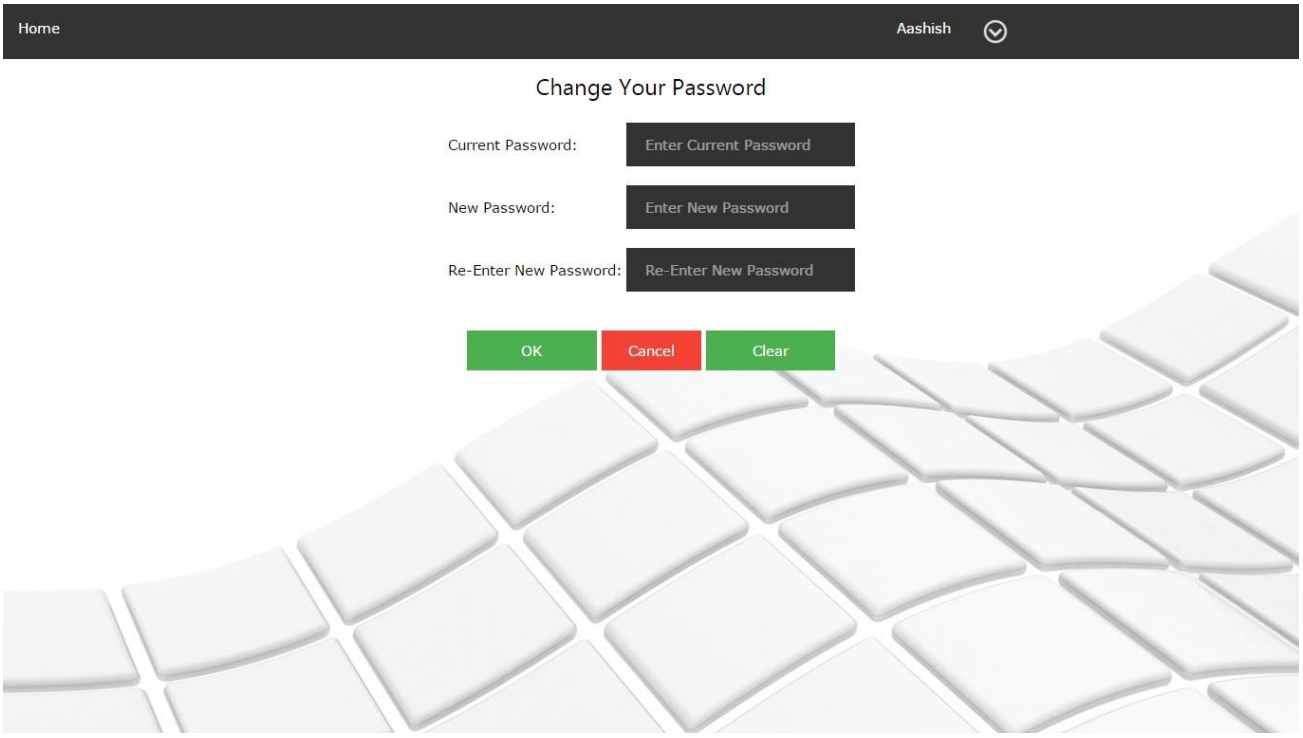
9) Update Profile



10) Settings



11) Change Password



12) Logout