**2. OVERALL DESCRIPTION**

**2.1 PROJECT PERSPECTIVE**

We have in mind a hybrid model proposed which is a mixture of Elliptical Curve Cryptography and a symmetric key algorithm. ECC is used to achieve the process of user’s verification and to keep the private data secure. AES (or whichever gives best results) algorithm is used which allows the user to store and access their data securely in the cloud by encrypting the data in the client side and decrypting the data after downloading from the cloud.

Since the private key is owned by the user of the data, no one can decrypt the data. Even though hackers can get the data through some approaches, they will not be able to access it. Moreover, user will securely authenticate itself by using different input parameters at the time of login to the cloud server. This scheme can make users assure about the security of data stored in the cloud. Here, we will apply an ECC and ECDH algorithm that provides same level of security as of other public key cryptosystems with less key size and strengthens the security of the algorithm.

Benefits are:

* Proper access mechanism to avoid unauthorised access to the information system
* Secure storage and access of data

The model would be hosted on a website. The website is designed keeping in mind an average user so that he can understand the functioning easily and does not have to hassle with the complex design of the website. The simple design of the website makes it very easy for any user to access it easily and make full use of the product.

**2.2 PROJECT FUNCTIONS**

This is the **landing page** of the web application – the very first page which is visible when the app is launched in the browser of the user’s system. This page lets the user know about the application and let the user to choose between first time registration and login features. Along with this, the page contains tabs for download and upload section also. It contains, app logo, tag line and makes the user aware about the goal of the application is short period of time and enhances aesthetic value.

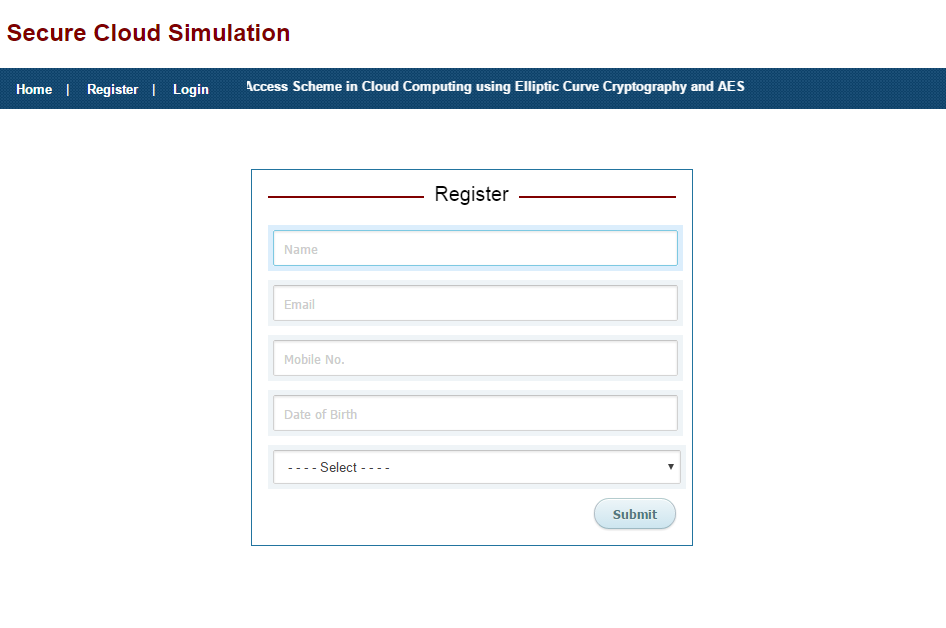


Fig. 1

This is the **Registration Page** of the application which is displayed when user is using the application for the first time or does not possess a user account.

The form here takes user’s full name, email address, mobile number, DOB and gender as input to store user’s information and validate the user before giving access to the app data. It first checks for all possible errors in the credentials on the client side itself using regular expressions and pattern matching. Later, if the data passes all the test cases, it is sent to server side to validate and store the details in the database to create a working profile of the user.

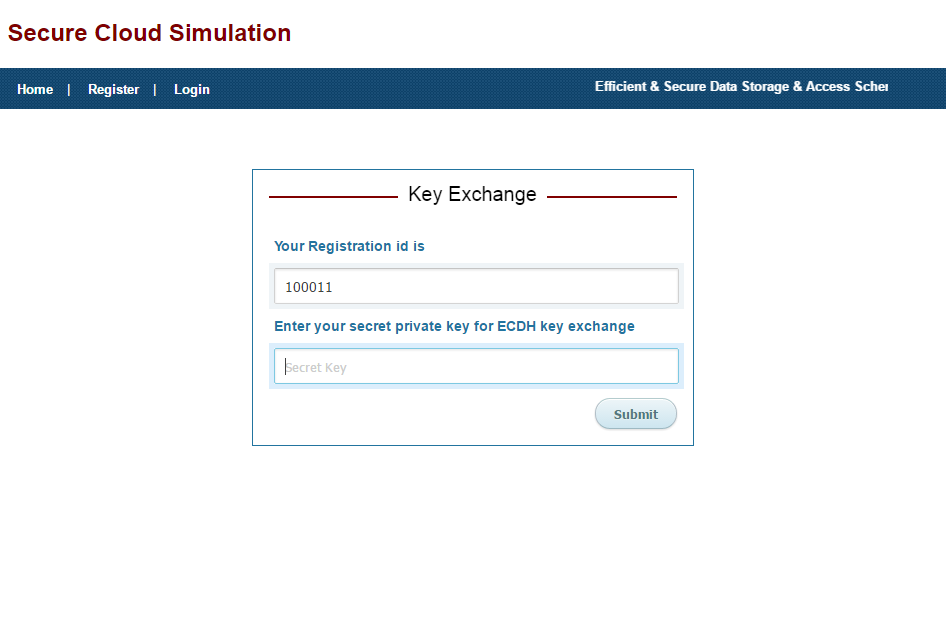


Fig. 2

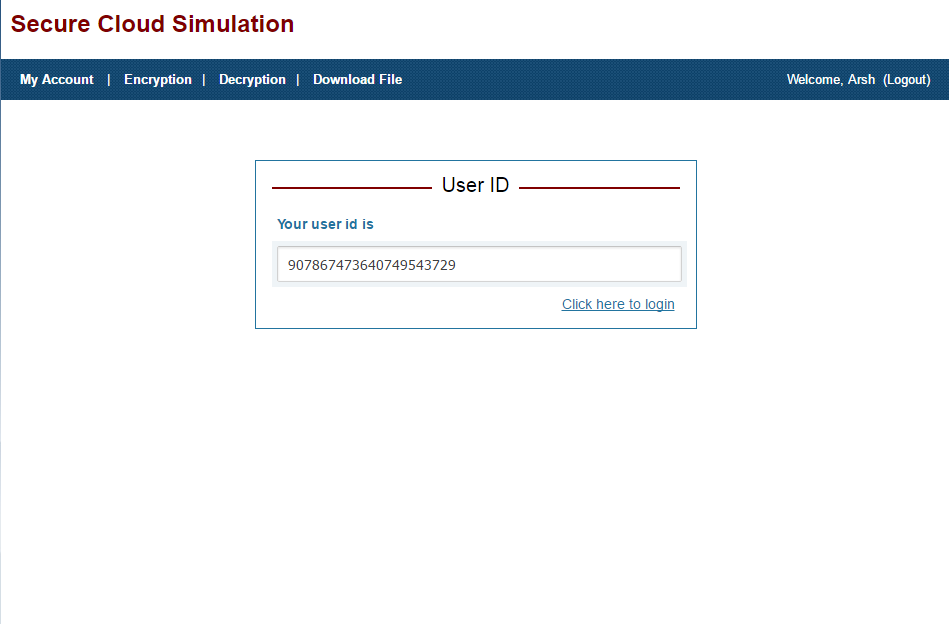
This is the **Key Exchange** page of the application. Using ECC algorithm, public key and private key both are generated. Here Sender will be used to encrypt the data and receiver i.e. B is used to decrypt the data by using its own private key. The form here takes, registration number along with secret key for successful key exchange.



Fig. 3

This is the 3rd stage of Registration process itself. Here ECDH key agreement has been automatically generated successfully and OTP (one time password) is sent to the email address given by the user on stage one of the registration. The app takes OTP as input and after proper validation of both the fields, it passes the control to the next stage of new user registration.

Fig. 4



On successful validation, the application generates a unique User ID. The user is requested to save this ID as, once the registration is done, the user will be using just the User ID generated here and the secret key entered earlier to use all the features and perform validation further in the application. The new user registration completes here and now the user can easily login to access his profile and use all the features of the application.

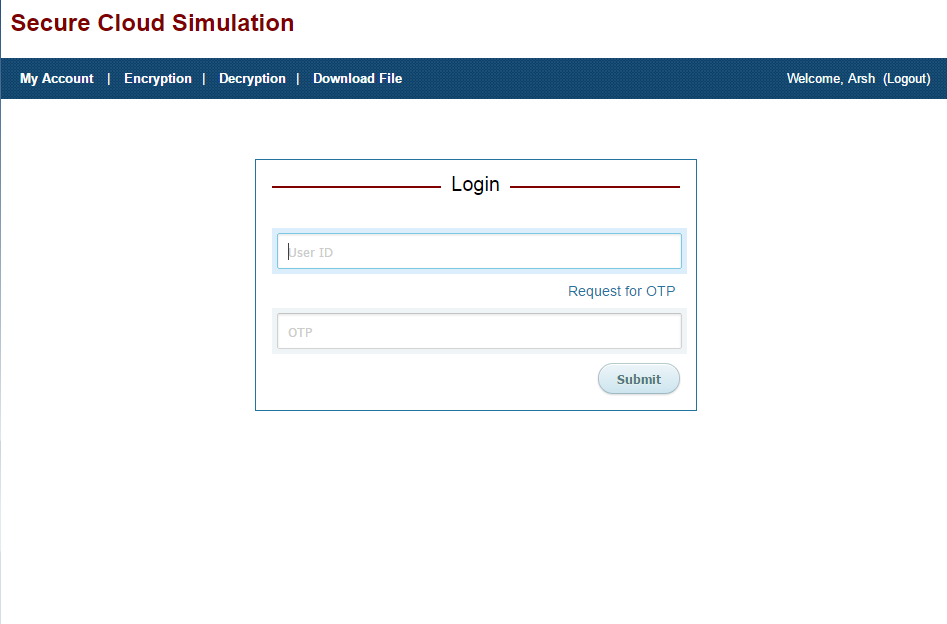


Fig. 5

This is the **Login page** of the application which is loaded when the user clicks on the login tab button. The page consists of a simple form which inputs the User ID of the user generated during registration process and the OTP which is immediately sent to the user’s email address as soon as he clicks on ‘Request for OTP’ button in the form. Later, both the values in the fields are validated over the server with the values in the database and if successful the user is taken to the next page that is Dashboard of the application, else error is generated.

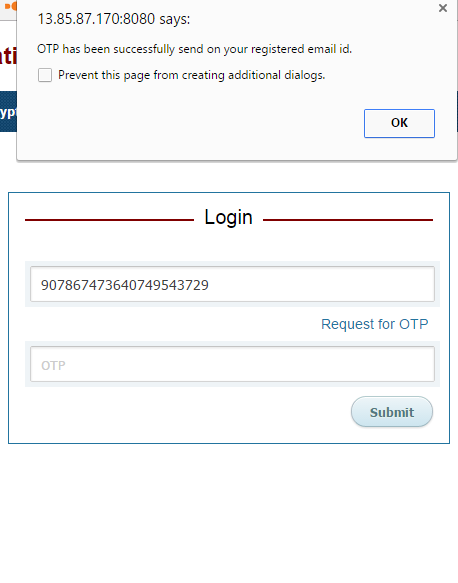


Fig. 6

Here a confirmation message is shown to verify that the OTP is successfully delivered to the registered email ID of the user.

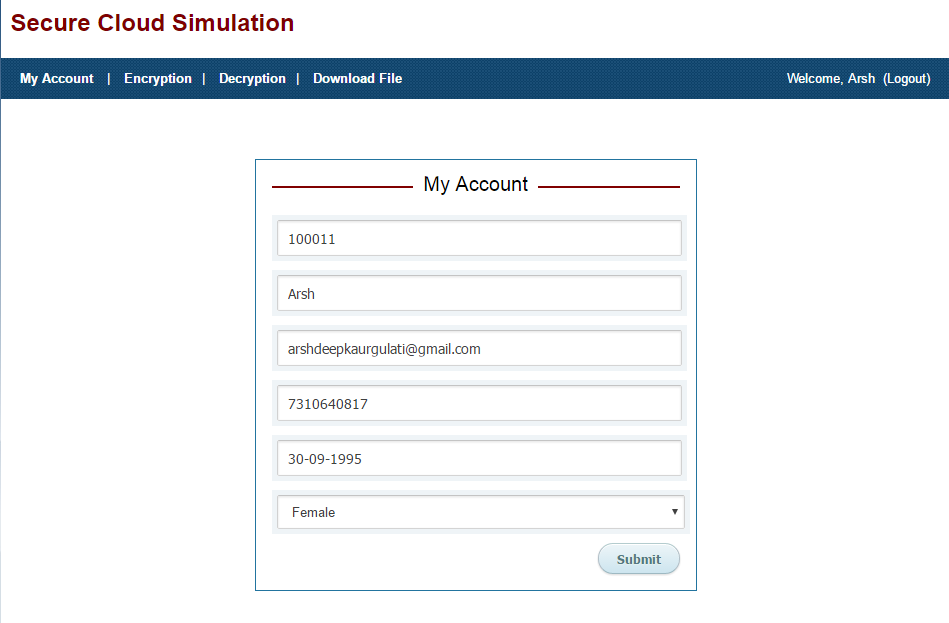


Fig. 7

This is the **Dashboard Page** of the application. When the user is able to successfully login into the app for the first time, he is taken to the My Account section where he is shown the current details of his account stored in the database and is given an opportunity to update any field if required and click on submit button to save. The dashboard page serves as a navigating page as from here the user can go-to encryption, decryption download and upload sections and can log out from the app if the work is done.

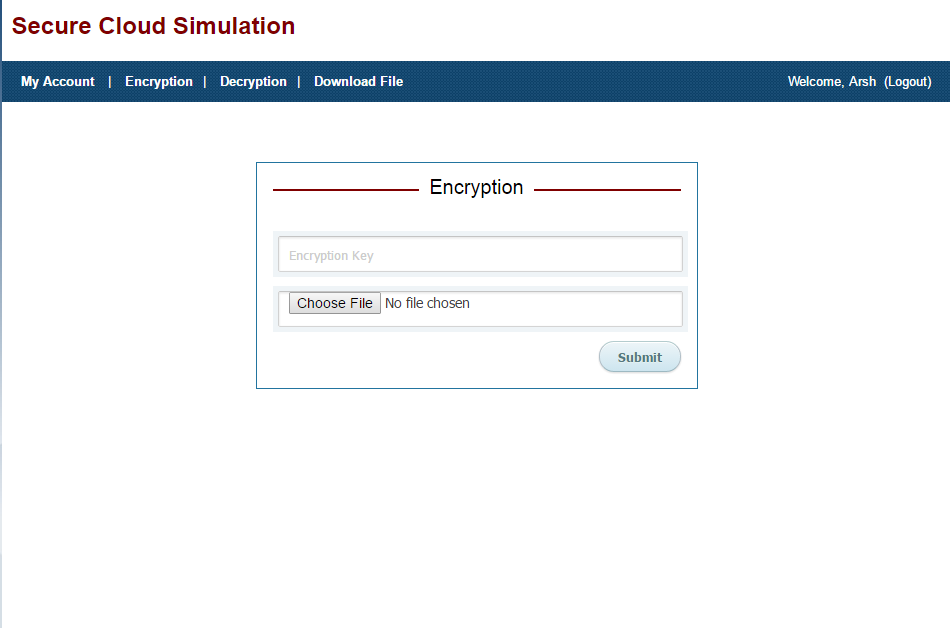


Fig. 8

This is the **File Encryption** page of the application. If the user wishes to upload a file in the cloud application and encrypt it to ensure security he would hit the encryption tab. Next he is asked to enter the Encryption key which is actually the same secret key entered at the time of registration also. After that, he is asked to select the file to be uploaded from his local system on which the application is running. Lastly, click on submit to process the encryption and saving of file.

Till now in this project we were able to complete till here, completing the modules like, 3 step registration, local database modelling and development, Key generation and OTP generation along with forms validations at each level, the complete login module and the user interface of the application. Currently, we are working on the file encryption and file handling module of the project, apart from this, we have to complete the file encryption-decryption keys, file secure downloading, implementation of AES algorithm and complete testing and load balancing of the application.

**2.3 CLASS TABLES AND DFD DIAGRAMS**

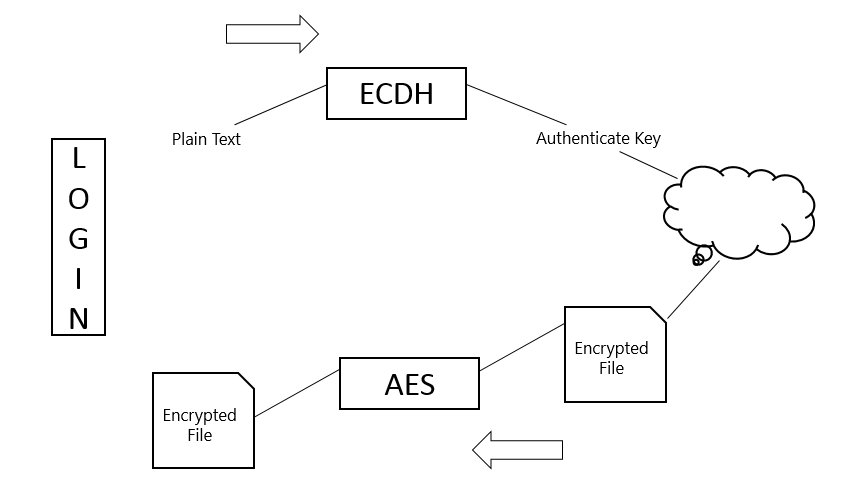


Fig. 17

Overall display of the project

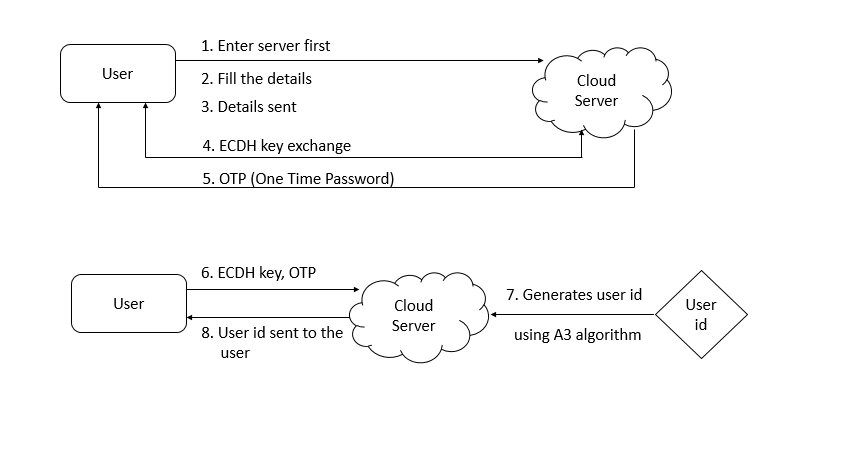


Fig. 18

Overall architecture with component description and dependency details

**2.3.1 DFD DIAGRAMS**

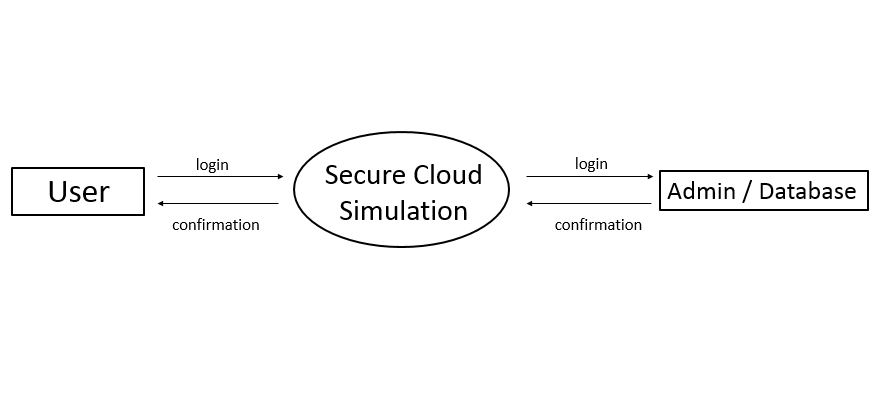


Fig. 19

Level-0 DFD

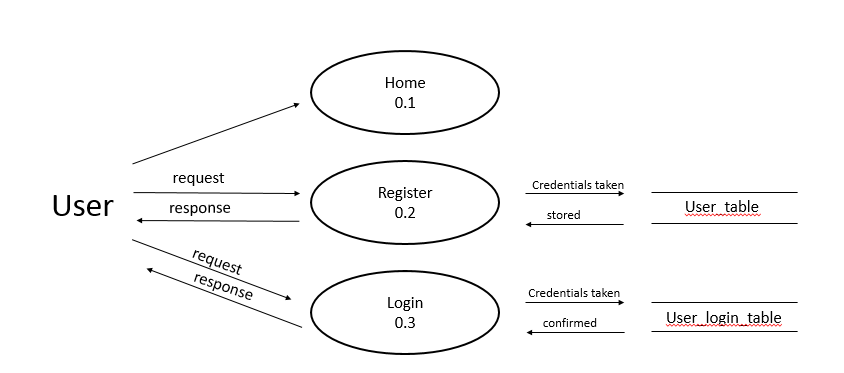


Fig. 20

Level-1 DFD

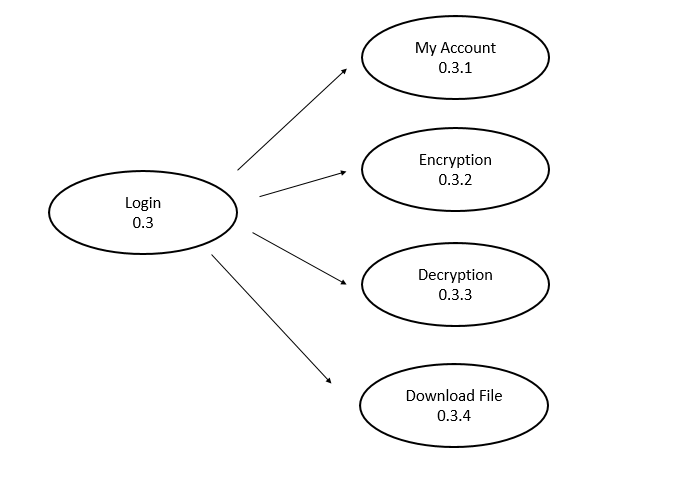
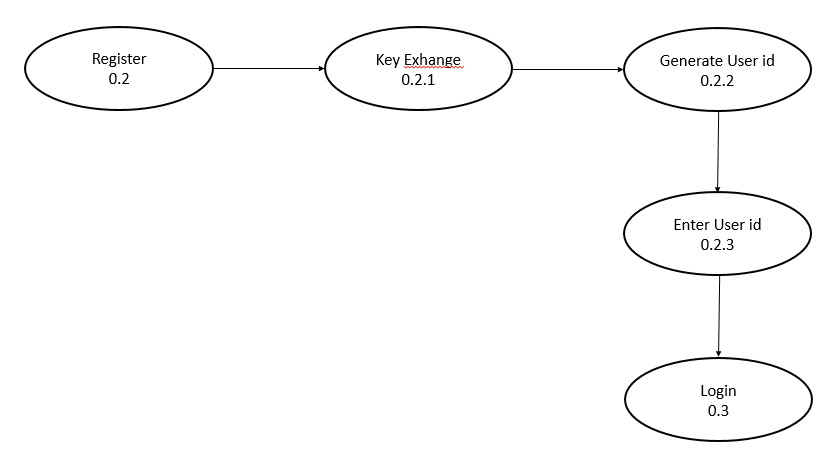


Fig. 21

Level-2 DFD

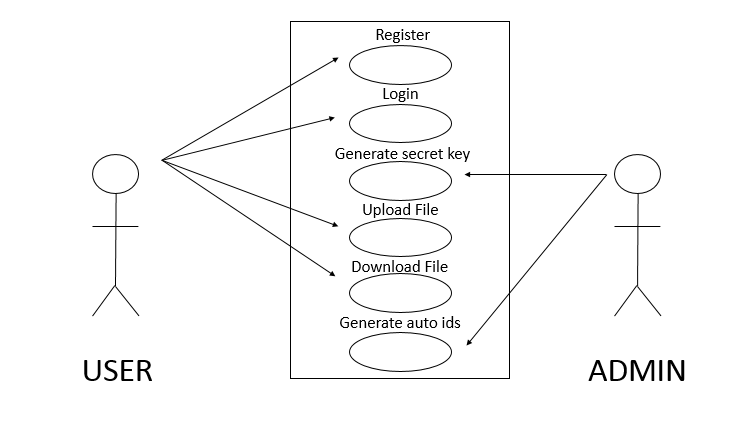


Fig. 22

User-Case Diagram

**2.3.2. CLASS TABLES**

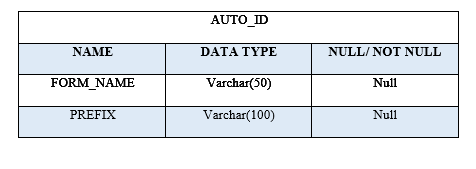


Fig. 23

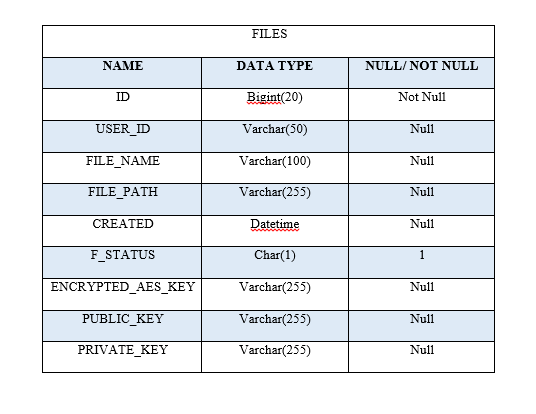


Fig. 24

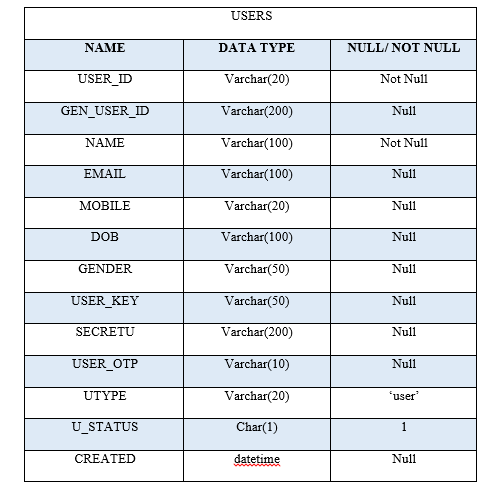


Fig. 25

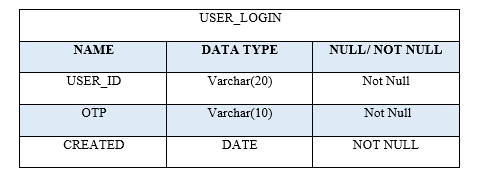


Fig. 26

**2.4 CONTRAINTS AND ASSUMPTIONS**

Following are the constraints of Secure Cloud Simulation:

* The software is only available on the web.
* The project implements a security model and shows the simulation.
* To use this model in real time a lot of modifications will be needed.
* The project will be tested against a certain set of test cases only.

Following are the assumptions:

* The user has good knowledge of operating a computer and web application.
* The computer system has internet and/or LAN connection enabled.
* It is mandatory for the internet to be turned on and active.
* The user device has enough memory available for installation and proper functioning of the application.