Practical - 2

Practical 2.1

Aim: To perform the system analysis task of your the system: Prepare Software Requirement Specification (SRS) Document.

INDEX

Password Manager

1. INTRODUCTION:

In this digital age, we tend to have a lot of online accounts on different platforms, each with a password we need to remember. On top of that, we are told to use “strong” passwords that are a combination of mixed case letters, numbers and symbols, use a password only once, and to change them all every three months. We are also told not to store passwords just anywhere because of the security risk. Now, it becomes extremely difficult to remember so many different complicated passwords. So what can be done to minimize the complexity while maintaining security? Well, there is a software solution to this problem. A ‘Password Manager’. A password manager is a software application that is used to store and manage the passwords that a user has for various online accounts. It requires creation of a “master” password that gives you access to your account. Some additional features such as automatic generation of strong passwords, 2-step authentication, automatic form filling feature and synchronization across all of your devices can also be included.

1.1 PURPOSE:

The purpose of the project is to store all of your passwords in one secure place where you would need to go through a 2-step verification process in order to gain access to your account and carry out certain operations. Thus, this app aims to eliminate the hassle of forgetting and resetting passwords and use of identical or weak passwords which could expose you to major risks.

1.2 SCOPE:

- This software will provide a web-browser-based graphical environment in which the users of the system will be able to perform various operations that are associated with storing, updating and retrieving Password information.
- The use will generate a master password at the time of registration which will grant him access of other stored passwords.
- Also, the user will have to provide an OTP sent to their registered device during registration, login and deletion of account.

1.3 Definition, Acronyms, Abbreviation:

- SQL - > Structured Query Language
- DFD - > Data Flow Diagram
- ER - > Entity Relationship
- IDE - > Integrated Development Environment
- SRS - > Software Requirement Specification

2. OVERALL DESCRIPTION

2.1 PRODUCT PRESPECTIVE:-

The proposed Password manager will take care of stored User names & Passwords. The user just have to input his master password and select a valid operation, which will be fulfilled by the system upon successful verification.

2.2 SOFTWARE REQUIREMENT

Front end:

- Web browser
- MS Visual Studio IDE
- A deployment platform (e.g. Heroku)

Back end:

- PostgreSQL

2.3 HARDWARE REQUIREMENT

- 2GB RAM
- 1.2 GHz processor
- Intel i3
- Windows / Mac OS / Linux / Android

2.4Assumptions and Dependencies:-

The product needs following third party applications for the development of the project:

- MS Visual Studio IDE for writing the source code.
- A web browser to test the source code’s output.
- A deployment platform.

3. SYSTEM FEATURES AND REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

- **R.1: Register**

  - Description : First the actual user will have to register/sign up. There are two different type of users.
  - The admin : He checks the payment status of users, validity of their account details and database operations.
  - The actual user : The user has to provide the required details regarding his account.

- **R.1.1: Sign up**

  - Input: Detail about the user as given on the sign up page, along with valid ID proof and complete payment for the using the services of the application.
  - Output: OTP to a provided email ID / phone no. and then confirmation of registration status and generation of master password if verified successfully.
  - Processing: All details will be checked and if any error are found then an error message is displayed else the master password will be generated.

- **R.1.2 : Login**

  - Input: Enter the user ID/email address and unique master password, along with OTP.
  - Output : User will be able to use the features of software.
  - Processing: If the entered password and OTP is correct, give access to the account features else give an error message.

- **R.2 : Manage stored usernames and Passwords.**

  - **R.2.1 : New entry**
    - Description:- Select or customize the most appropriate category where the password will be stored.
    - Add the portal address and desired username corresponding to the password to be generated.

- Processing: If the entered password is correct give access to add a new record else give an error message.

- **R.2.2 : Retrieve password**
  - Input: Search the website / portal name by keywords or by category. Enter the master password.
  - Output: - Show the User name and password associated with that portal.
  - Processing: If the entered password is correct, give access to retrieve the record else give an error message.

- **R.2.3 : Update Password**
  - Input: Search or navigate to the desired portal entry. Enter the master password. Manually change the data.
  - Output: Display the changed data.
  - Processing: If the entered password is correct, give access to update the record else give an error message.

3.2 Non Functional Requirements

- **Usability Requirement**

The system shall allow the users to access the system from the phone or a computer using a web browser application like Google Chrome, Mozilla Firefox, Opera browser, Internet Explorer, Safari and several others. The system uses a web application as an interface. The User Interface shall be kept minimal and self-explanatory. Still, if the user needs any help, there is a help option also available. The system shall be user friendly and hence easy to use.

- **Availability Requirement**

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- **Efficiency Requirement**

  Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

- **Accuracy**

  The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.

- **Performance Requirement**

  The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

- **Reliability Requirement**

  The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.

3.3 USER CHARACTERSTICS

We have 2 levels of users :

- User module:
  In the user module User can:
  - Register and do payment
  - Login
  - Create / Modify Categories
  - New Entry
  - Search Record
  - Update Record
  - Delete Record
  - Give Feedback

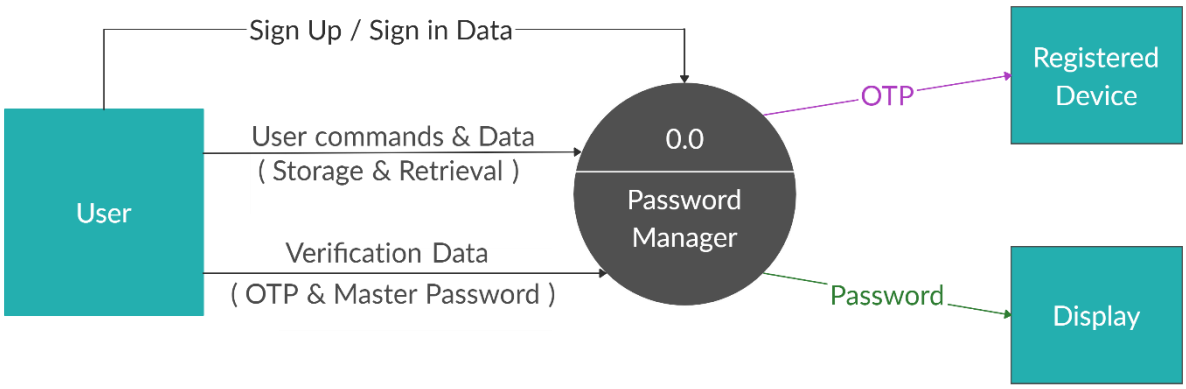
- Administrator module:
  In the admin module Admin can:
  - Check payment status
  - Check the disclosable details and validity of the records inserted by client
  - Do maintenance of Application
  - Check the Feedback of Clients

3.4 CONSTRAINTS:

The user should have provided valid identity proof along with full payment.

3.5 DATA FLOW DIAGRAM:

3.5.1 Context diagram (Level 0 DFD)



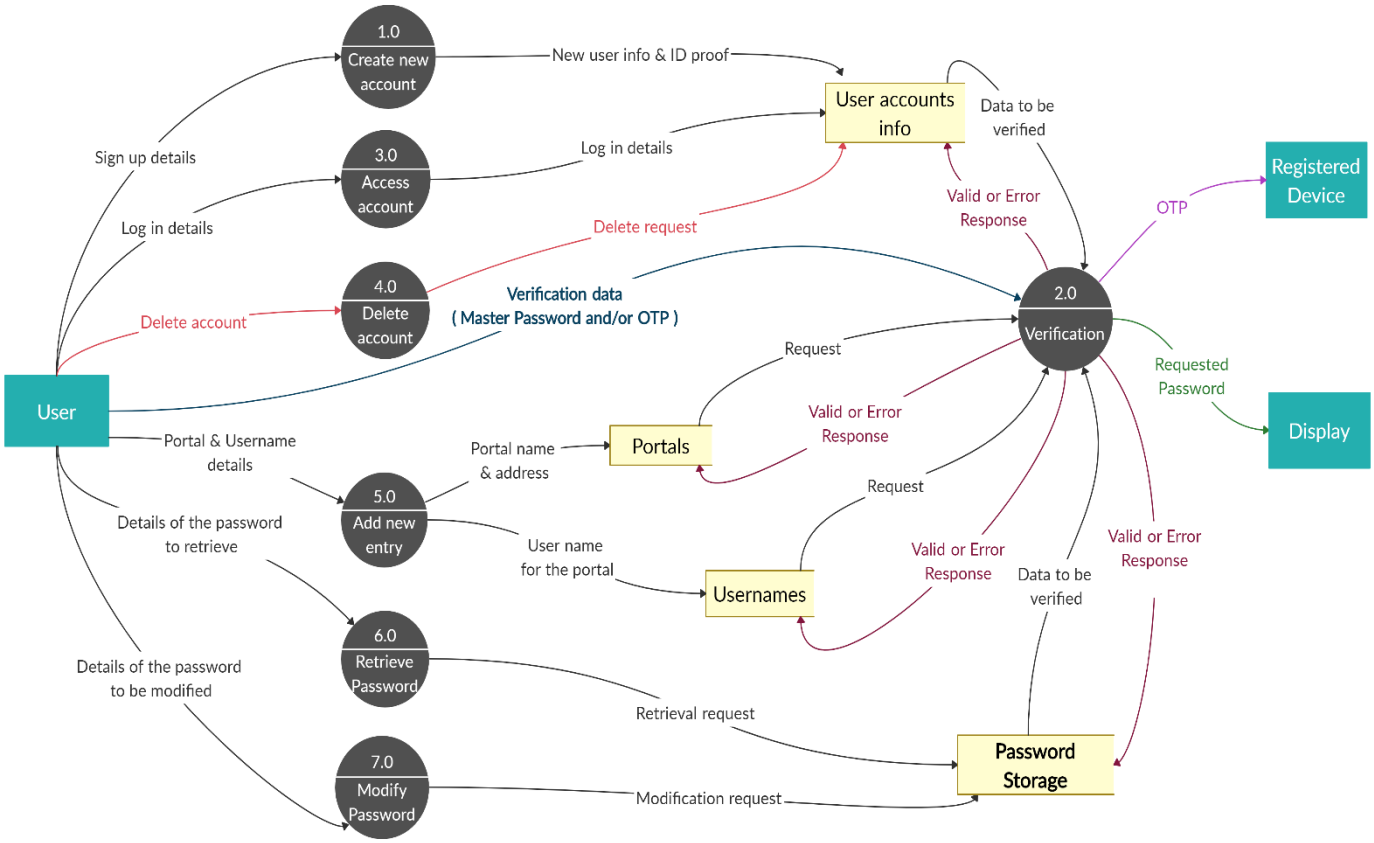
[Fig. 1 - Context diagram (Level 0 DFD)]

- The **Password Manager** is itself the main process in the context diagram. It takes input from the external entity *User* in the form of *Sign Up/ Sign in data* during registration and while accessing account, respectively.

- *User* also feeds as input, the desired operations to perform after gaining access, such as adding a new entry to store password, retrieving password as well as other valid operations.

- The main process then sends as output, an OTP to the *Registered Device* entity as programmed and if the verification data (Master Password and/or OTP) entered by user verifies, only then is the user able to gain permission to access his sensitive data through the entity *Display*. All the data are represented by labelled arrows.

3.5.2 Level 1 DFD:



[Fig. 2 - Level 1 DFD)]

- The main process (i.e. the system) of level 0 is decomposed into a total of 7 processes in the level 1 diagram.

- As the process names suggest, the *User* is able to:
    - Create a new account (Registration)
    - Access his existing account (Log in)
    - Delete his account
    - Add a new entry for storing password
    - Modify password if needed
    - Retrieve password

- The remaining process, marked as 2.0 in the diagram, is the *Verification* process which is executed as a **requirement** along with all of the processes listed. Thus, this process demands the needed verification data from the *User* which is shown by the labelled arrow.

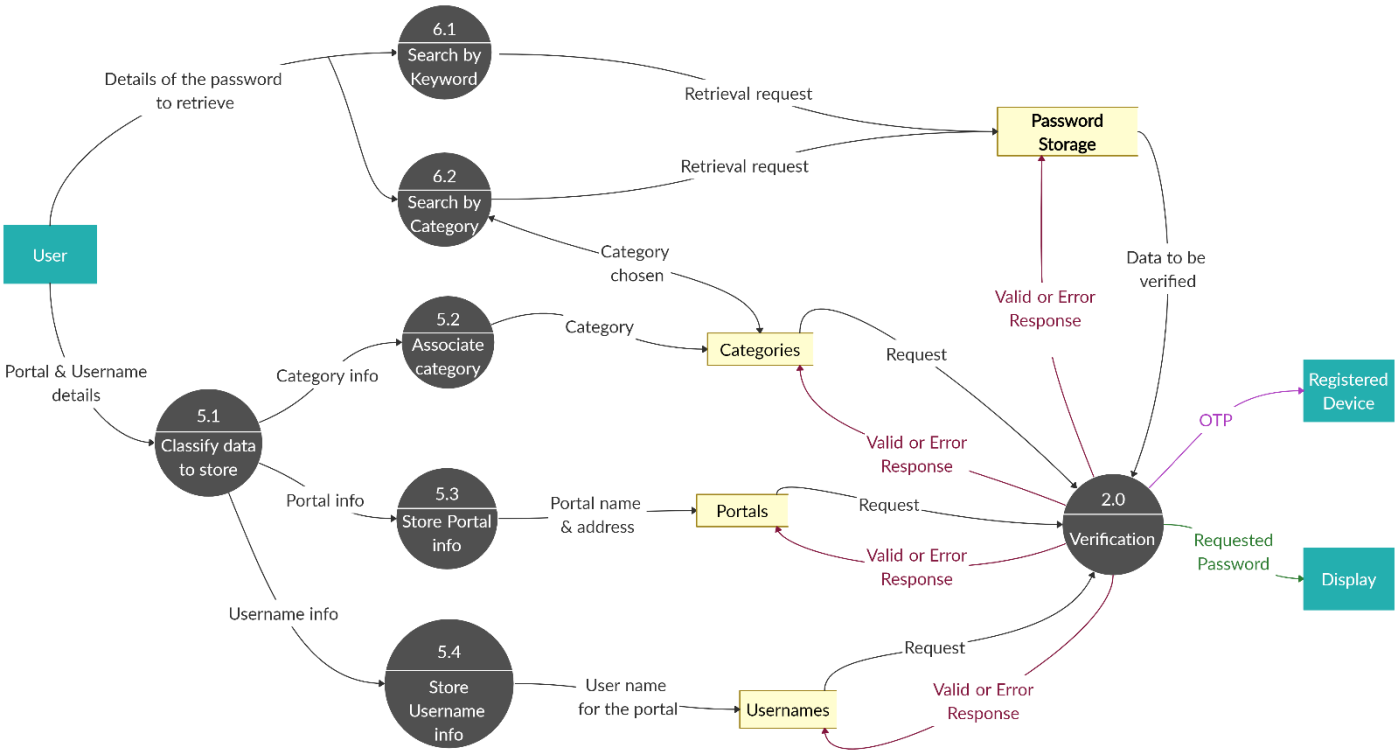
- While processes like Registration, Log in and Delete account **require both Master Password and OTP** (which consequently triggers the OTP output to the registered device), the other processes, after logging in, ask only for the Master Password.

- Level 1 diagram also features 4 data stores:
    - *User accounts info* – for storing and sending for verification, the info received during registration and login process and while removing data during the deletion of account.
    - *Portals* – for storing the details like web addresses of various platforms on which the user holds an account.
    - *Usernames* – for storing the username / user ID or email ID pertaining to a portal.
    - *Password Storage* – the place where important sensitive data is housed.

- Upon receiving requests to meddle with the stored data, the referenced data is sent for *Verification* which returns the appropriate valid or error response to decide further procedure.

- In this way, the level 1 DFD gives us the big picture and explains the interrelation and flow of data between the members.

3.5.3 Level 2 DFD:



[Fig. 3 - Level 2 DFD)]

- Level 2 DFD expands upon one or more processes and provides us with an under the hood view of the working. Here, the processes *5.0 Create a new entry* and *6.0 Retrieve Password* from level 1 are further decomposed, as the main task of a password manager is to store and retrieve the desired password whenever needed.

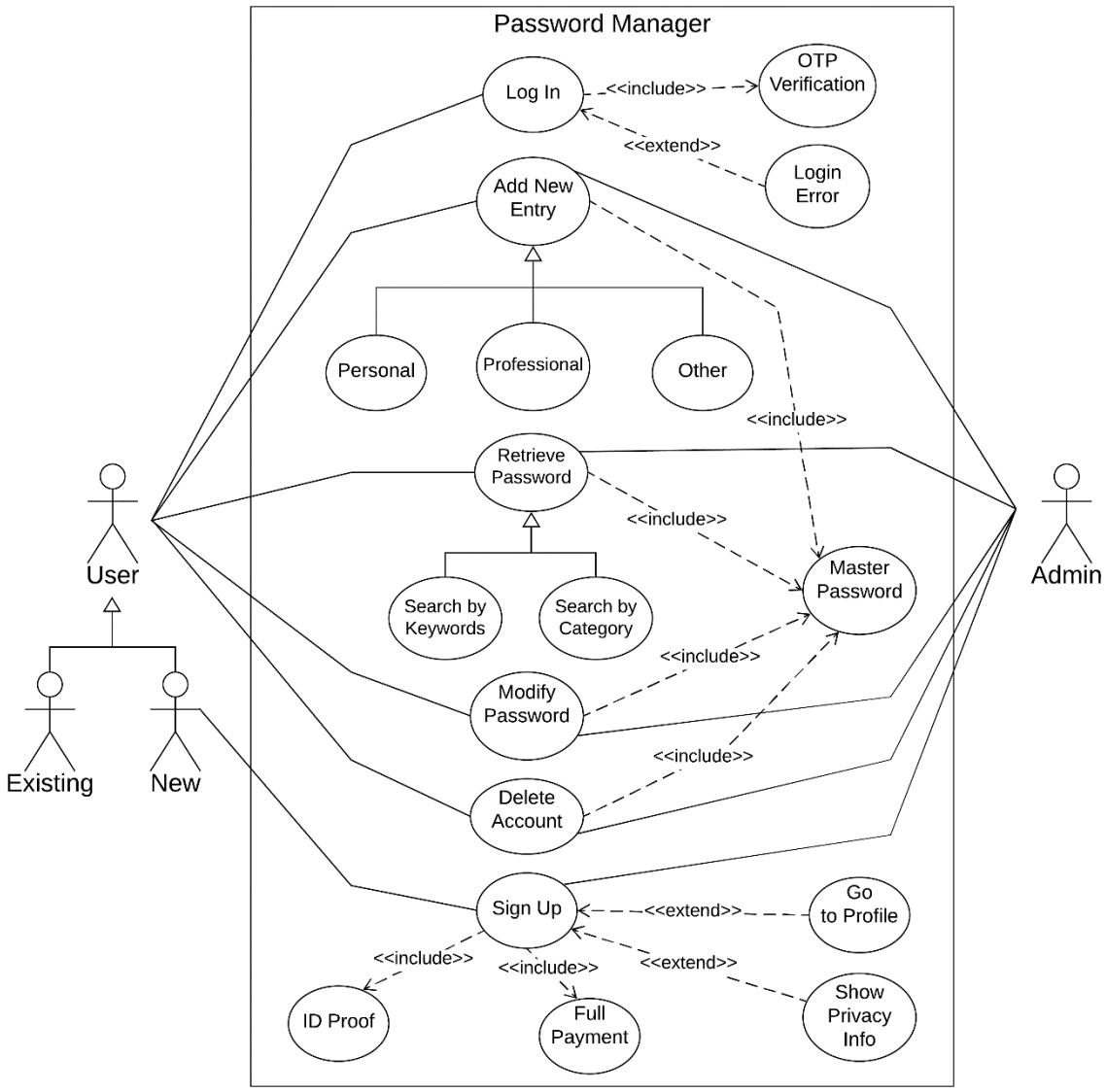
- The process of storing password for a new portal can be bifurcated into sub-processes like *associating a category* (personal, professional, custom) using the *Categories* data store, and storing the corresponding portal and username details into their respective data stores.

- The retrieval process is backed by methods such as *searching a password by keywords* or *searching by category*, which makes the organization of password details a bit more neat. The category search process also sends and receive data from the *Categories* data store.
- And naturally, these processes require a green signal from the *Verification* process to perform the function successfully.

➢ Thus, the data flow diagram helps system designers and other stakeholders during initial analysis stages to visualize a current system or one that may be necessary to meet new requirements.

3.6 USE CASE

3.6.1 USE CASE DIAGRAM:



3.6.2 USE CASE MODEL DESCRIPTION:

The Use-Case diagram concerning a password manager consists of two actors:

1. **User**: The actual end customers who use the software. This actor is a generalization of the following two actors:
    a. **Existing User**: The registered users who are already using the app.
    b. **New User**: Those users that have yet to register and create an account.

2. **Admin:** The one who handles the backend side of the website and aids the user in using the features provided by the application.

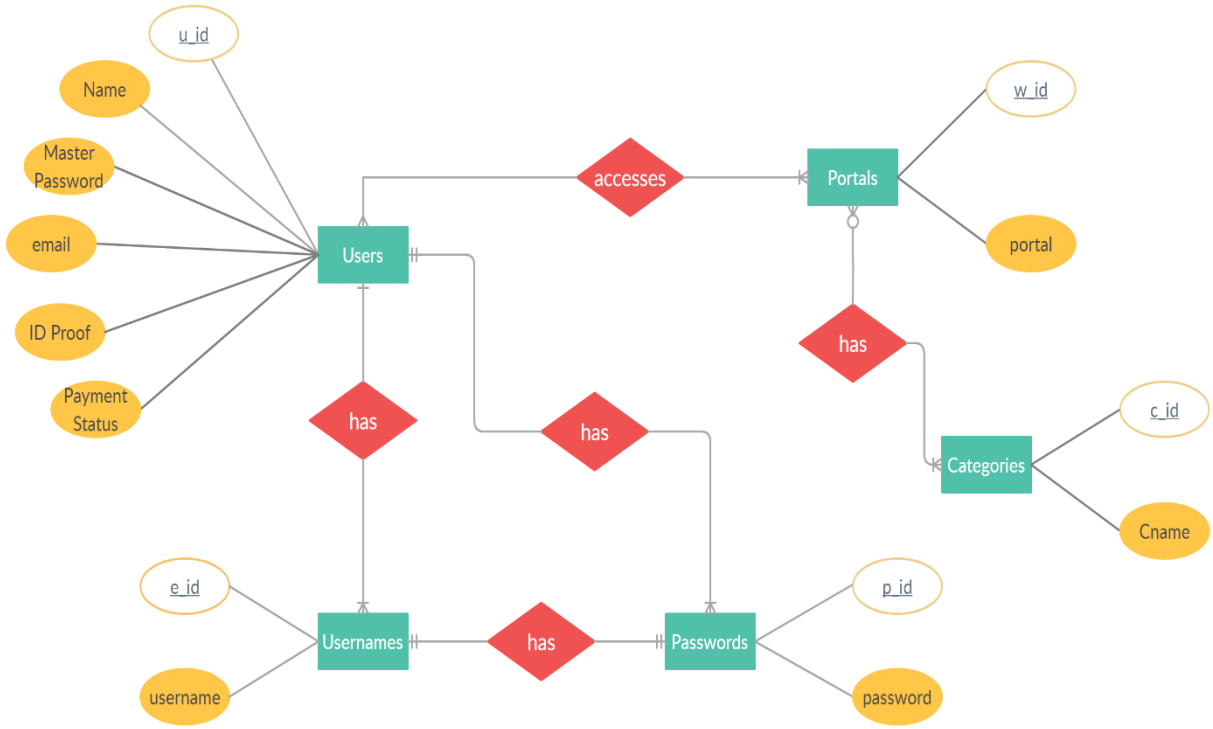
User actor:

- User actor is the one who makes use of the app's features. New users are connected to the app by sign up (registration).
- For Existing users, Login is essential to get access to his account.
- As shown in the diagram, a user can log into his account by OTP verification which possess an include relationship with the Log in feature.
- Occasionally, the user gets a login error if the OTP mismatches.
- After successful login, a user can add a new entry in terms of a portal or website by choosing the category he seems appropriate.
- User can also retrieve password by searching with the help of keywords or by browsing or navigating through the categories.
- If the user wishes to discontinue his use of the software, he can do so by accessing the Delete Account feature and providing 2-step verification to terminate his membership. This will result in elimination of all of his data from the app's database.
- It is noteworthy that the above three features require the input of the user's unique master password for their successful completion.
- The feature specially for New users is the sign up / register module which offers help for setting up the profile or for knowing in detail, the privacy information of the app.
- The new registration step, however, requires the user to provide his valid identity proof and full payment for using the app's services.

Admin actor:

- The main job of Admin is to regulate and manage the user's requests.
- Admin is connected to the database and controls the functions of adding a new user, adding a new entry, associating a category with an entry, providing the user his queried password after successful verification and wipe out the user's data in case the user desires to do so.
- All of this behind the scene functionality is handled by Admin.

3.7 ER diagram:



The above shown figure depicts the Entity-Relationship diagram pertaining to a password manager. It comprises 5 relations among 5 entities namely Users, Portals, Categories, Usernames and Passwords.

The Users entity mainly stores the data recorded during the registration of various individuals that use the software. It consists of attributes such as u_id (the primary key) representing the unique no. associated with each user to distinguish one from the others. The Name, email and ID Proof attributes store the user's name, email ID and identity proof respectively. The Payment Status is a Boolean attribute that shows whether the user has paid the user fee in full or not. Finally, the Master Password attribute stores the master code that the user is required to generate at the time of registration which is essential in accessing their other passwords and sensitive information. The Users entity possesses couple of 'has' relationships with the Usernames and Passwords entities and 'accesses' relationship with the Portals entity since users have access to portals and their categories, their usernames and naturally the associated passwords.

The Portals entity stores all of the user's added portals for which he has saved usernames and strong passwords in the password manager. It has the w_id (the primary key) and portal attributes to store and identify the user's portals. Another entity which is closely associated with Portals is the Categories entity having c_id (the primary key) and Cname attributes which

consists of several categories such as personal, professional or any custom-created category. They both possess a ‘has’ relationship since each Portal can be classified into some or the other category.

The Usernames (primary key - e_id) and Passwords (primary key - p_id) entities contain the most important data in their corresponding attributes, username and password. They share a ‘has’ relationship among them since each username for a portal has one and only one password linked to it.