

# **CHAPTER 1**

## **INTRODUCTION**

When an online service suffers a data breach - there's a risk that an intruder will discover your password and gain access to your account. That danger is multiplied if the compromised password has been used across multiple sites.

Passwords present an online dilemma; seemingly every service we use online requires a password, and for those passwords to be secure, they have to be complex. However, unless we are blessed with a powerful memory, it's impossible to remember half a dozen mixed-case, alphanumeric, special-character inclusive, lengthy random keys – so it's no surprise that people resort to reusing passwords.

This is where password managers come in; they do the remembering for us.

### **1.1 OVERVIEW**

Our project provides the user an easy to use interface where he can lock up all his passwords and retrieve it whenever required. Instead of having to remember possibly hundreds of passwords you only need to remember one. Prior to using a password manager most people simply wrote down their passwords. Whether it is in a book or stored in plain text file on their computer. Unfortunately, there's always the risk that prying eyes can come across what should be private. The advantage which password wallet offers is that our data is stored in an encrypted format. That way even if our device is lost or stolen our passwords cannot be read without knowing the master password.

### **1.2 TYPES OF PASSWORD MANAGERS**

There are 2 main types of password managers – password managers that store the passwords in the Cloud and those that store our passwords locally on our computer. The advantage of storing your passwords remotely “in the cloud” is that all your devices will pull your passwords from one centralized location. The downside to cloud based

password managers is a perceived risk by some that your information could be compromised by a hacker. Therefore to avoid such security issues, the best solution is to store our passwords in an encrypted format locally on our pc, smart phone, or tablet.

### **1.3 DESCRIPTION OF PASSWORD WALLET**

Password Wallet is a software application that helps a user to store and organize passwords. It stores the passwords in an encrypted format, requiring the user to create a master password; a single, ideally very strong password which grants the user access to their entire password database. The master password is stored in a hashed form in the database using SHA-256 algorithm. The user passwords are encrypted and stored in the database using AES algorithm which can later be decrypted using the key provided by the user. While the core functionality of this application is to securely store large collections of passwords, additional features for modifying existing passwords, deleting unwanted passwords and guidelines for usage of the application are provided.

The main functionalities provided by the application to the user are-

- Add passwords
- Modify passwords
- Delete passwords

Therefore, if we have trouble remembering our passwords, or are simply tired of clicking the forget password button, then Password Wallet is the application which solves all problems by providing effective solutions.

### **1.4 IMPLEMENTATION OF AES ENCRYPTION AND SHA-256**

AES (acronym of Advanced Encryption Standard) is a symmetric encryption algorithm. The algorithm was developed by two Belgian cryptographer Joan Daemen and Vincent Rijmen. AES uses a block length of 128 bits and a key length that can be 128, 192, or 256 bits. The longer the key, the more effective is the security. If there is ever a break in AES that reduces the effective number of operations required for cracking it, a bigger key gives us a better chance of staying secure.

### 1.4.1 Strength of AES Encryption

For cryptographers, a cryptographic "break" is anything faster than a brute force—performing one trial decryption for each key. This includes results that are infeasible with current technology. The largest successful publicly known brute force attack against any block-cipher encryption was against a 64-bit RC5 key by distributed.net in 2006. AES has a fairly simple algebraic description. In 2002, a theoretical attack, termed the "XSL attack", was announced, purporting to show a weakness in the AES algorithm due to its simple description. Since then, other papers have shown that the attack as originally presented is unworkable.

Key size	Time to Crack
56-bit	399 seconds
128-bit	$1.02 \times 10^{18}$ years
192-bit	$1.872 \times 10^{37}$ years
256-bit	$3.31 \times 10^{56}$ years

Figure 1: Time to crack Cryptographic Key versus Key size

As shown above, even with a supercomputer, it would take 1 billion billion years to crack the 128-bit AES key using brute force attack. This is more than the age of the universe (13.75 billion years). If one were to assume that a computing system existed that could recover a DES key in a second, it would still take that same machine approximately 149 trillion years to crack a 128-bit AES key.

There are more interesting examples. The following snippet is a snapshot of one the technical papers from Seagate titled "*128-bit versus 256-bit AES encryption*" to explain why 128-bit AES is sufficient to meet future needs.

*If you assume:*

- *Every person on the planet owns 10 computers.*
- *There are 7 billion people on the planet.*
- *Each of these computers can test 1 billion key combinations per second.*
- *On average, you can crack the key after testing 50% of the possibilities.*

*Then the earth's population can crack one encryption key in  
77,000,000,000,000,000,000,000 years!*

### 1.4.2 Description of SHA-256

SHA stands for Secure Hash Algorithm. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their one-way nature: given a computed hash value, it is very difficult to derive the original data.

Some of the applications that use cryptographic hashes, such as password storage, are only minimally affected by a collision attack. Constructing a password that works for a given account requires a pre-image attack, as well as access to the hash of the original password (typically in the shadow file) which may or may not be trivial. Reversing password encryption (e.g., to obtain a password to try against a user's account elsewhere) is not made possible by the attacks.

## **CHAPTER 2**

### **SOFTWARE REQUIREMENT SPECIFICATION**

#### **2.1 INTRODUCTION**

The introduction provides an overview of the entire SRS with purpose, scope, definitions, and references. The SRS aims to provide guidelines for how the project will be deemed successful and how these concepts and ideas will be created effectively. It provides a detailed overview of our software product, its parameters and goals.

##### **2.1.1 Purpose**

The purpose of this document is to gather, analyse and given an in-depth insight of Password Wallet which is a password manager that helps us to manage our passwords in a secure way. We can put all your passwords in one database, which is locked with one master key. This application provides all required functionalities to the users for adding, modifying and deleting passwords in an interactive manner.

##### **2.1.2 Document Convention**

The following document conventions have been used to ensure the easy of readability

- Font: Times New Roman
- Main Headings- Font size 16, Bold
- Sub-Headings- Font size 14, Bold
- Sub-sub-headings- Font size 12, Initial Capital letters

##### **2.1.2.1 Definitions**

- Button- A user interface element that allows user to click and inform the system to take an action
- Checkbox- A user interface element that allows a user to inform the system that they have selected a particular item.
- User- The person who operate the software product.

- Title- The website or URL for which the password is stored.
- GUI- Graphical User Interface
- Open Source Software- Software for which the code is freely available for use and research.
- API- Application Programming Interface
- JDK- Java Development Kit

### **2.1.3 Project Scope**

The scope of the project “Password Wallet” is to provide users an option of integrating the features of flexibility and security into their password storage. Our application makes it very easy for users to maintain multiple accounts having different passwords without the concern of having to remember all those passwords. It aims at replacing the traditional alternative of writing down all passwords where security is a major concern. Therefore our strong encryption and hashing techniques takes care of the security issues password maintenance.

### **2.1.4 References**

- [1] JAVA Complete Reference, Seventh Edition by Herbert Schildt.
- [2] Cryptography and Network Security, B.A. Forouzan, D.Mukhopadhyay, 2<sup>nd</sup> Edition
- [3] Network Security Essentials (Applications and Standards) by William Stallings
- [4] Core Java by Dr R. Nageswara Rao
- [5] [http://en.wikipedia.org/wiki/Password\\_manager](http://en.wikipedia.org/wiki/Password_manager)
- [6] <http://bradleyandcompany.com/password-managers/>
- [7] UML Diagrams- <http://staruml.io/>

## **2.2 OVERALL DESCRIPTION**

The description of the product includes a discussion about its features, types of users to whom the application may be useful, the environment in which the application can be operated etc.

### **2.2.1 Product Perspective**

Password Wallet is a self-contained product which can help users and organizations from diverse realms in managing their passwords which may either be passwords used by the organization or online passwords. Every user has to sign up by creating an account initially and later log in to use all the functionalities of this application.

### **2.2.2 Product Features**

The main function of Password Wallet is to allow its users a secure storage and retrieval of their passwords.

The user can add any number of passwords to his account which gets encrypted and stored in the database. He can then modify those passwords which also get changed in the database or even delete them permanently. The passwords can be viewed at any time by the user by giving the appropriated title.

### **2.2.3 User Classes and Characteristics**

- Open Source Community

The open source community is expected to be the main user class of this application. Nowadays, with the dawn of social networking site, online study forums and communities, passwords have become an entry card for every site. Thus, our application comes handy to these users.

- Programmers

The next class of users is programmers who are people who constantly surf the net and are expected to use this product. Programmers have to deal with multiple accounts and sites and hence will have to maintain multiple passwords.

- **Business Organisations**

The multi-national companies and business enterprises are another class of users who need to maintain large number of passwords. Our application is of great use in such organisations where every department has to manage their passwords.

#### **2.2.4 Operating Environment**

Operating System- Windows 95/98/2000/XP/8/8.1

Password wallet needs JDK to be installed on the system and MS Access is usually present every system as it is present in the MICROSOFT package.

#### **2.2.5 Design and Implementation Constraints**

In the time which was available to us, we have tried to implement all possible security features in the product. In our database, the password fields are encrypted. As an added security, the whole database can be encrypted including User ID fields.

### **2.3 SYSTEM FEATURES**

This section of the SRS describes all the functionalities that the product provides with a detailed instructions guide to the users of how to use those features.

#### **2.3.1 CREATE ACCOUNT**

Every user has to initial create his or her personalized user account.

- **Description and Priority**

There is a form that the user has to fill up which consists of his user-name and password. He can then log into his account using this username and password.

Priority-High Priority

- **Stimulus/Response Sequence**

Launch Application → Click Sign Up → Fill the form → Submit



### 2.3.2 LOGIN

This is the step where the user enters into his account by entering his credentials.

- **Description and Priority**

The user is asked to enter his username and password. If the entered fields match the fields in the database then the user is given access to his account. Else, he has to re-enter his credentials.

Priority- High priority

- **Stimulus/Response Sequence**

Launch Application → Enter Username and Password → Click Login

### 2.3.3 ADDING PASSWORDS

This feature allows users to add passwords to his account.

- **Description and Priority**

After logging into his account the user can add any number of passwords and each password should have an associated title with it. These passwords can later be viewed, modified or deleted by the user.

Priority- Medium priority

- **Stimulus/Response Sequences**

Login → Click Manage → Click Add Password → Fill the details → Click Submit

### 2.3.4 MODIFYING PASSWORDS

The user can replace existing password with new passwords.

- **Description and Priority**

This is an important feature as every user needs to constantly change his passwords. This can be done by modify password button. The old password is replaced by the new password.

Priority- Medium priority

- **Stimulus/Response Sequences**

Login→Click Manage→Click Modify Password→Select Title →Click Submit

### 2.3.5 DELETING PASSWORDS

The user can remove existing passwords once he doesn't need them.

- **Description and Priority**

Once a user does not need a password anymore, he can delete that password from the database. This is also a necessary feature as users create a lot of temporary accounts. Once the need of the account is over, the password is also useless.

Priority- Low priority

- **Stimulus/Response Sequences**

Login→ Click Manage→ Click Delete Password→ Select Title → Click Delete

### 2.3.6 VIEWING PASSWORDS

The user can retrieve the stored passwords.

- **Description and Priority**

The user can view his password anytime by selecting the appropriate title. These titles are names of the sites or accounts of which the user wants to retrieve the password.

Priority- High priority

- **Stimulus/Response Sequences**

Login → Click View → Select Title → Click View Password

### 2.3.7 HELP

It launches a user manual which guides the user and helps in resolving doubts related to the application.

- **Description and Priority**

This feature is a user guide which gives a brief description of how to use the application. The Help button is present on every screen for the user.

Priority- Medium priority

- **Stimulus/Response Sequences**

Click Help button at the bottom of the screen and follow the instructions.

## **CHAPTER 3**

### **ANALYSIS**

#### **3.1 EXISTING SYSTEM**

- Most of the online accounts require passwords which should include letters, digits and special characters in order to increase its complexity.
- In order to remember passwords belonging to multiple accounts, people tend to write down passwords on paper which makes it completely defenceless.
- Once a password is forgotten, the user has to go through a series of steps which is time-consuming. Also, certain accounts do not facilitate password retrieval, which makes it necessary to create a new account.

#### **3.2 DRAWBACKS OF THE EXISTING SYSTEM**

- With the numerous online accounts that a person maintains, it becomes impossible to remember all the passwords.
- Once an intruder gains access to any password, then he gets access to all accounts on which that particular password has been used by the user.
- There is no mechanism using which a user can update his passwords regularly as constant change in passwords might cause the user to get confused with his own passwords.

#### **3.3 PROPOSED SYSTEM**

- The proposed system is an easy-to-use safe which stores all the passwords of a user in a secure manner.
- The program stores the passwords in a highly encrypted database.
- SHA-256 is used to hash the master password which unlocks the wallet. SHA-256 is a 256-bit cryptographically secure one-way hash function. In

contrast to many other hashing algorithms, no attacks are known yet against SHA-256.

- The passwords are encrypted using AES algorithm. Both of these ciphers are regarded as extremely secure.

### 3.4 MERITS OF THE PROPOSED SYSTEM

- **Strong security**—Password Wallet uses AES encryption to encrypt its password databases and SHA-256 password hash for the master password.
- **Portable**—Password Wallet is portable, carry it on a USB stick and run it.
- **Easy Database Transfer**—A password database, containing a single file, is easy to transfer between computers.
- **Multiple user keys**—Use a master password to decrypt the complete database or carry a key file.
- **Graphical User Interface**—Password Wallet is an easy-to-use application for users owing to its interactive interface which is highly understandable.

### 3.5 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation-

- Technical feasibility
- Economic feasibility
- Social feasibility

#### 3.5.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Is the underlying technology used secure and up to date with the current trend?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate responses to inquiries?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Password Wallet uses SHA-256 to hash the master password of the application and all the user passwords are encrypted using AES encryption algorithm. It uses JDBC and Microsoft Access .accdB ODBC Driver. By way of illustration: Cracking a 256 bit AES key with a state-of-the-art supercomputer would take longer than the presumed age of the universe.

Password Wallet prevents any discrepancy in the passwords and maintains data integrity. The initial step is where the user has to enter the master password after which he is given access into the application. Just by giving the keyword of the site or account of which the user wants the password, he can retrieve the password easily.

### **3.5.2 Economic Feasibility**

A system can be developed technically and that if installed must be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefits must equal or exceed the costs. This application has no maintenance costs and its development cost is very less compared to the service which it provides to the users.

### **3.5.3 Social Feasibility**

The aspect of this study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level

of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### **3.6 SCOPE OF THE PROJECT**

This application can be used in business organisations and enterprises where multiple password usage is a norm and its protection has to be given top priority. Having a separate, lengthy, and random password for every site is considered the best way to secure data. Password Wallet keeps every username and password pair in an encrypted database, protected by a single master password or key (the only one you have to remember). Our application is a must-use for every user who wants to enjoy online services without the concern of a password compromise. Also, with the increased password attacks it becomes necessary that the users regularly change their passwords. In such a case password managers modify option is a great help as all that the user has to do is set a new password and not worry about remembering it.

## **CHAPTER 4**

### **REQUIREMENTS**

#### **4.1 SOFTWARE REQUIREMENTS**

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices.

The software requirement specifications for developing this project are as follows-

- Operating System- Windows 95/98/2000/XP/8/8.1
- Front End- Java
- Back End- MS Access
- IDE- Notepad++

##### **4.1.1 Java Platform**

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

##### **4.1.2 Java Development Kit**

The Java Development Kit (JDK) is an implementation which includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). The JDK is a development environment for building applications, applets, and components using the Java programming language. The JDK



includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

#### **4.1.3 MS Access Database**

Microsoft Access, also known as Microsoft Office Access, is a database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. Like other Office applications, Access is supported by Visual Basic for Applications (VBA), an object-oriented programming language that can reference a variety of objects including DAO (Data Access Objects), ActiveX Data Objects, and many other ActiveX components. Visual objects used in forms and reports expose their methods and properties in the VBA programming environment, and VBA code modules may declare and call Windows operating-system functions. We have used MS Access database as the repository to store all the passwords and using commands can access them when needed by the user.

### **4.2 HARDWARE REQUIREMENTS**

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware.

The minimum hardware requirement specifications for developing this project are as follows-

- Processor- Pentium-3 or later versions.
- Speed- 1 Ghz or faster 32-bit or 64-bit processor.
- RAM- 256 MB
- Floppy Drive- 1.44 MB
- Monitor- 15 VGA colour
- Keyboard- 110 keys enhanced.

## CHAPTER 5

### DESIGN

#### 5.1 UML DIAGRAMS

UML is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems.

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard.

- UML stands for Unified Modeling Language.
- UML is different from the other common programming languages like C++, Java, COBOL etc.
- UML is a pictorial language used to make software blue prints.

So UML can be described as a general purpose visual Modeling language to visualize, specify, construct and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well like process flow in a manufacturing unit etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization UML is become an OMG (Object Management Group) standard.

UML can be described as the successor of object oriented analysis and design.

### 5.1.1 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

So the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

#### How to draw Class Diagram?

Class diagrams are the most popular UML diagrams used for construction of software applications. So it is very important to learn the drawing procedure of class diagram. Class diagrams have lot of properties to consider while drawing but here the diagram will be considered from a top level view.

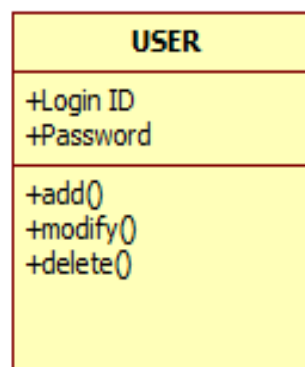


Fig 5.1 Class Diagram for User

### 5.1.2 Usecase Diagram

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and Statechart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

#### How to draw Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analysed the functionalities are captured in use cases.

So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw an use case diagram we should have the following items identified.

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

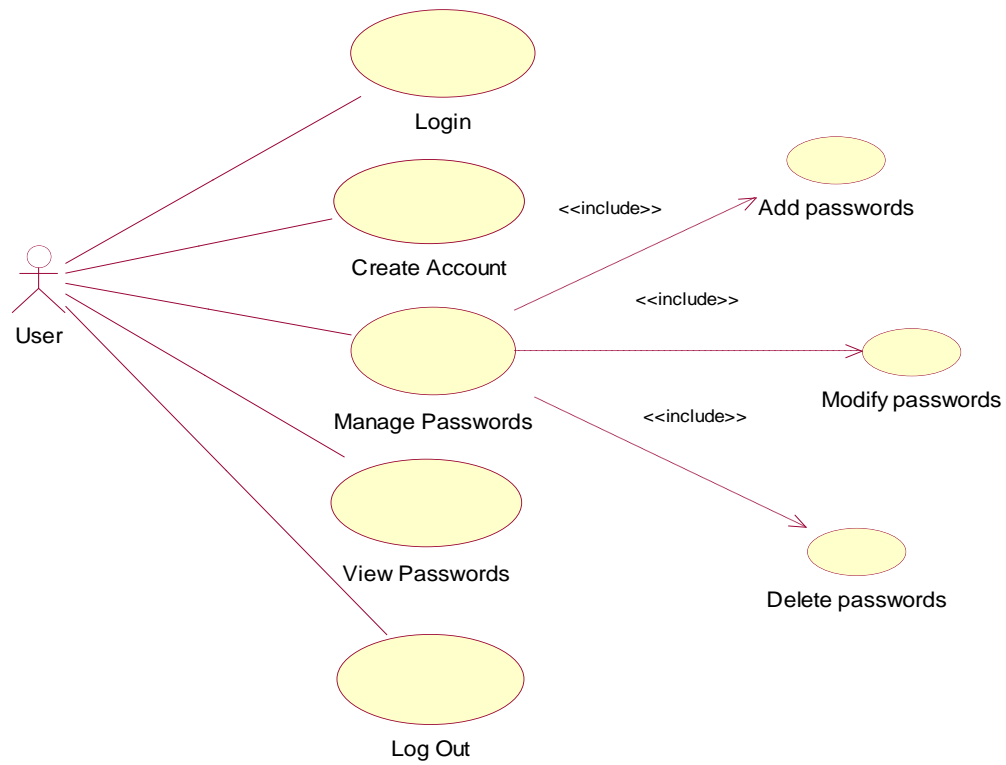


Fig 5.2 Usecase Diagram of Password Wallet

### 5.1.3 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

So the purposes can be described as-

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.

Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram are not exactly a flow chart as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane etc.

Before drawing an activity diagram we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities we need to understand how they are associated with constraints and conditions.

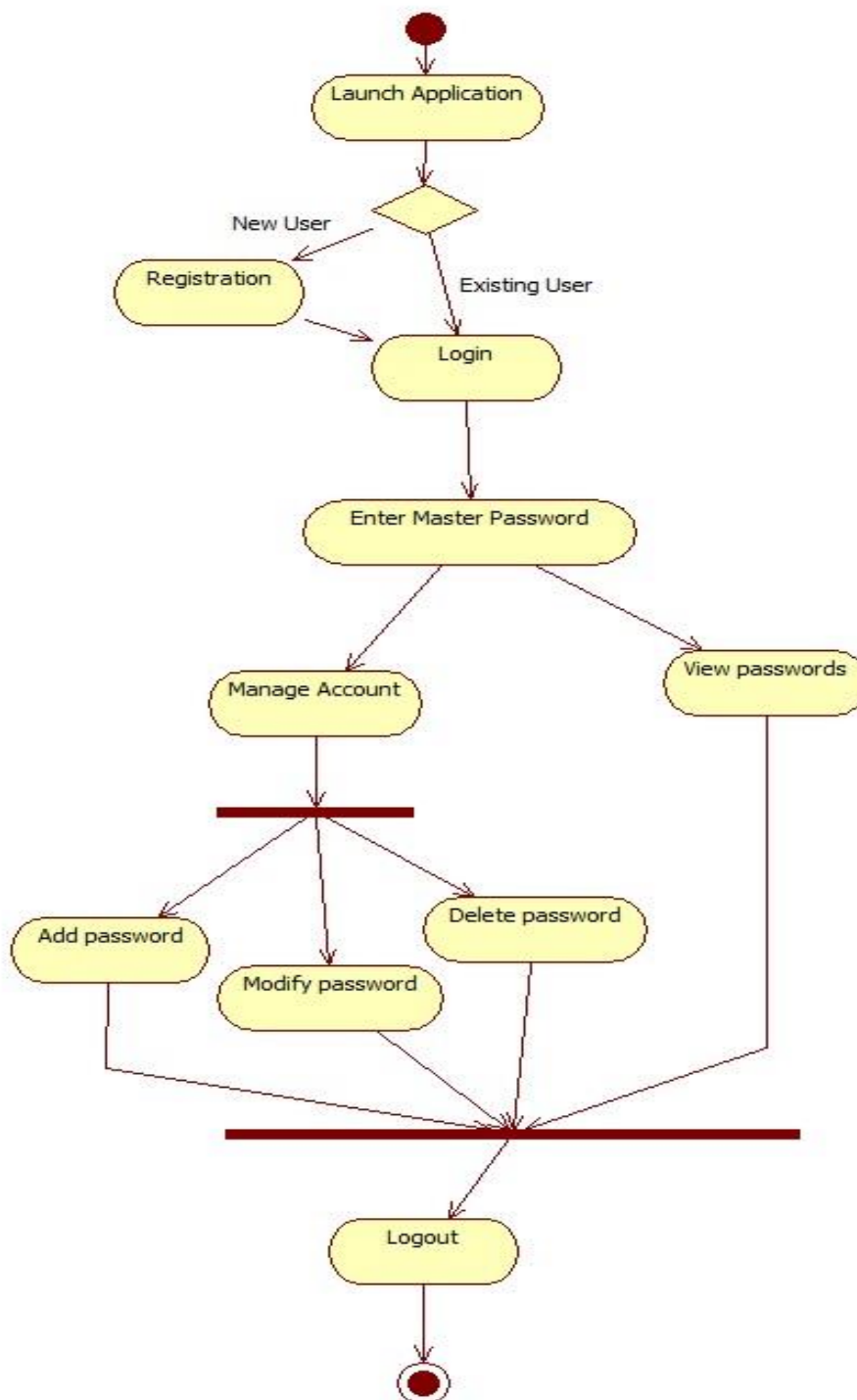


Fig 5.3 Activity Diagram of Password Wallet

### 5.1.4 Sequence Diagram

The diagram is used to describe some type of interactions among the different elements in the model. So this interaction is a part of dynamic behaviour of the system.

The purposes of interaction diagrams are to visualize the interactive behavior of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

So the purposes of interaction diagram can be describes as:

- To capture dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe structural organization of the objects.
- To describe interaction among objects.

So to capture the dynamic aspect we need to understand what a dynamic aspect is and how it is visualized. Dynamic aspect can be defined as the snap shot of the running system at a particular moment. We have two types of interaction diagrams in UML. One is sequence diagram and the other is a collaboration diagram. The sequence diagram captures the time sequence of message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.



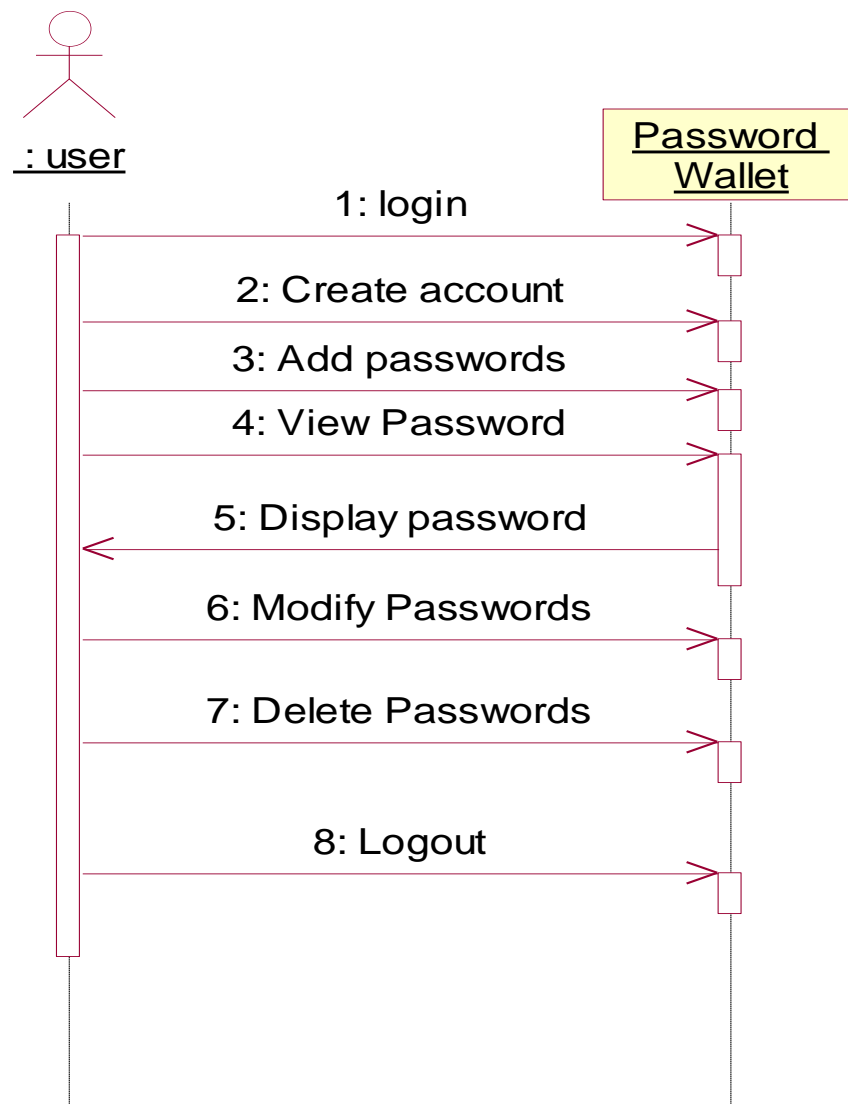


Fig 5.4 Sequence Diagram of Password Wallet

### 5.1.5 Statechart Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A state chart diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime. And these states are changed by events. So statechart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram is used to describe the states of different objects in its life cycle. So the emphasis is given on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

Statechart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

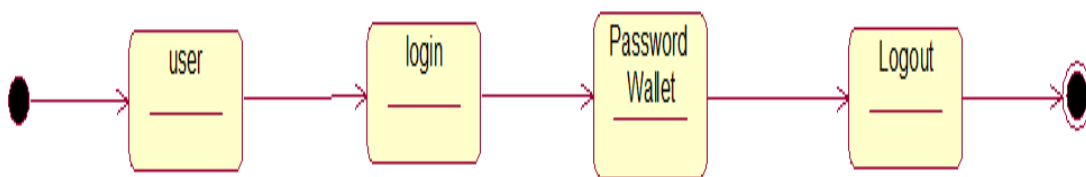


Fig 5.5 Statechart Diagram for Password Wallet

### 5.1.6 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

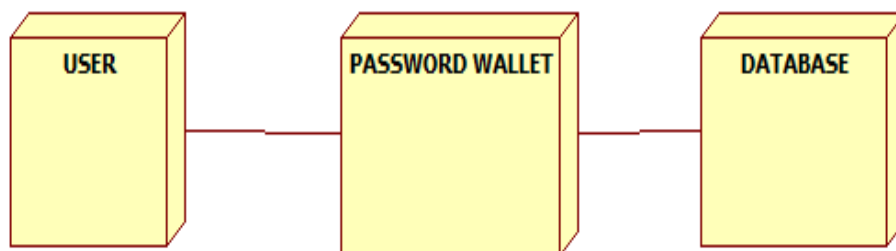


Fig 5.6 Deployment Diagram of Password Wallet

## 5.2 DATABASE TABLES

### 5.2.1 MAIN\_TB Table

NAME	DATATYPE
UID	VARCHAR(30)
PWD	VARCHAR(120)
SEC_QN	INTEGER(25)
SEC_AN	VARCHAR(25)

### 5.2.2 UID Table

NAME	DATATYPE
TITLE	VARCHAR(50)
URL	VARCHAR(30)
UID	VARCHAR(30)
PWD	VARCHAR(120)

## CHAPTER 6

### IMPLEMENTATION

#### 6.1 MODULES AND DESCRIPTION

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project. They are

- **End user Training:**

The successful implementation of the new system will purely upon the involvement of the officers working in that department. The officers will be imparted the necessary training on the new technology

- **End User Education:**

The education of the end user start after the implementation and testing is over. When the system is found to be more difficult to understand and complex, more effort is put to educate the end used to make them aware of the system, giving them lectures about the new system and providing them necessary documents and materials about how the system can do this.

- **Training of application software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

- **Post Implementation View:**

The department is planning a method to know the states of the past implementation process. For that regular meeting will be arranged by the concerned officers about the implementation problem and success.

This application comprises of a single important module.

**Administrator-** The administrator or user will have control over the entire system. He can add/modify details which includes file-name, password, URL address etc. A single password is maintained for the whole database, which is known only to the admin. Only after entering this password, would the admin be able to gain access to the list of passwords.

## 6.2 TECHNOLOGIES USED

### 6.2.1 Java Database Connectivity

A JDBC driver is a software component enabling a Java application to interact with a database. JDBC drivers are analogous to ODBC drivers, ADO.NET data providers, and OLE DB providers.

Java Database Connectivity (JDBC) is a front-end tool for connecting to a server to ODBC in that respect, However JDBC can connect only Java clients and it uses ODBC for the connectivity. JDBC is essentially a low-level application programming interface. It is called a low-level API since any data manipulation, storage and retrieval has to be done by the program itself. Some tools which provide a higher-level abstraction or expected shortly.

The next question that needs to be answered is why we need JDBC, once we have ODBC on hand. We can use the same ODBC to connect the entire database and ODBC is a proven technology. Problem for doing this is ODBC gives a 'C' language API, which uses pointers extensively. Since Java does not have any pointers and is object-oriented sun Microsystems, inventor of Java developed to suit its needs.

### 6.2.2 Requirements to use JDBC

To use JDBC you need a basic knowledge of database and SQL. Apart from this you need the jdk1.1 (Java Development Kit 1.1) or a version of Java since jdk1.1 and above come bundled with JDBC software.

After that you need to have a back-end database engine for which a JDBC driver is available. When JDBC drivers are not available JDBC-ODBC bridge drivers are used to access the database through ODBC. Back-end is not need when JDBC driver is capable of storing and retrieving the data itself, or if JDBC-ODBC bridge and the ODBC driver can be store and retrieve the information.

### 6.2.3 JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

- **JDBC-ODBC bridge plus ODBC driver:** The Java Soft bridge product provides JDBC access via ODBC drivers. Note that ODBC binary code and in many cases database client code must be loaded on each client machine that uses this driver. As a result, this kind of driver is most appropriate on a corporate network where client installations are not a major problem, or for application server code written in Java in three-tier architecture.

- **Native-API partly-Java driver:** This kind of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

- **JDBC-Net all-Java driver:** This driver translates JDBC calls into a DBMS-independent net protocol that is then translated to a DBMS protocol by server. This net server middle ware is able to connect its all-Java clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC alternative. It is likely that all vendors of this solution will provide products suitable for Internet use. In order for these products to also support Internet access, they must handle the additional requirements for security, access through firewalls, etc., that

the Web imposes. Several vendors are adding JDBC drivers to their existing database middle ware products.

- **Native-protocol all-Java driver:** This kind of driver converts JDBC calls into the network protocol used by DBMS directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Internet access. Since many of these protocols are proprietary, the database vendors themselves will be the primary source. Several database vendors have these in progress.

Eventually, we expect the last two drivers will be preferred way to access database from JDBC. And the first two driver categories are interim solutions where direct all-Java drivers are not yet available. The last driver is in some sense the ideal one. However, there are many cases where JDBC-Net all-Java driver may be preferable. For example, where a thin DBMS- independent client is desired, or if a DBMS-independent protocol is standardized and implemented directly by many DBMS vendors.

#### 6.2.4 Microsoft Access ODBC Driver

This driver was first released with Office 2007. It is possible to use the Microsoft Access .accdB ODBC Driver to connect to old .mdb (Access 97-2003) files as well. The driver is available in both 32 bit and 64 bit versions.

##### 6.2.4.1 Setting up an ODBC Data Source For MS Access

- Click **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**. The ODBC Data Source Administrator window appears.
- Click the **System DSN** tab, and click **Add** to create a new data source. The Create New Data Source window appears.
- Select **Microsoft Access Driver (\*.mdb)** and click **Finish**. The ODBC Microsoft Access Setup window appears.
- Enter the Data Source Name. In this example we created a data source called **TestChannel**.
- **(Optional)** Enter a description of your data source.



- Under Database, click the **Create** button to create your database. The New Database window appears.
- Enter the Database Name (e.g. TestChannel.mdb).
- Choose where to save the database and click **OK**. A dialog window appears, confirming you have successfully created your database.
- Click **OK** to complete the data source setup. The newly created database appears under System Data Sources.

### 6.2.5 Implementation of SHA-256 Hashing and AES Encryption Algorithm

A hash is not ‘encryption’ – it cannot be decrypted back to the original text (it is a ‘one-way’ cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare ‘hashed’ versions of texts, as opposed to decrypting the text to obtain the original version. SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available. While SHA-1 has not been compromised in real-world conditions, SHA-256 is not much more complex to code, and has not yet been compromised in any way. The 256-bit key makes it a good partner-function for AES.

As first publicly accessible, from the NSA for the classification "top secret" approved cipher, the Advanced Encryption Standard (AES) is one of the most frequently used and most secure encryption algorithms available today. The algorithm is based on several substitutions, permutations and linear transformations, each executed on data blocks of 16 byte – therefore the term block cipher. Those operations are repeated several times, called “rounds”. During each round, a unique round key is calculated out of the encryption key, and incorporated in the calculations. Due to the block structure of AES, the change of a single bit either in the key or in the plaintext block results in a completely different cipher text block which is a clear advantage over traditional stream ciphers.

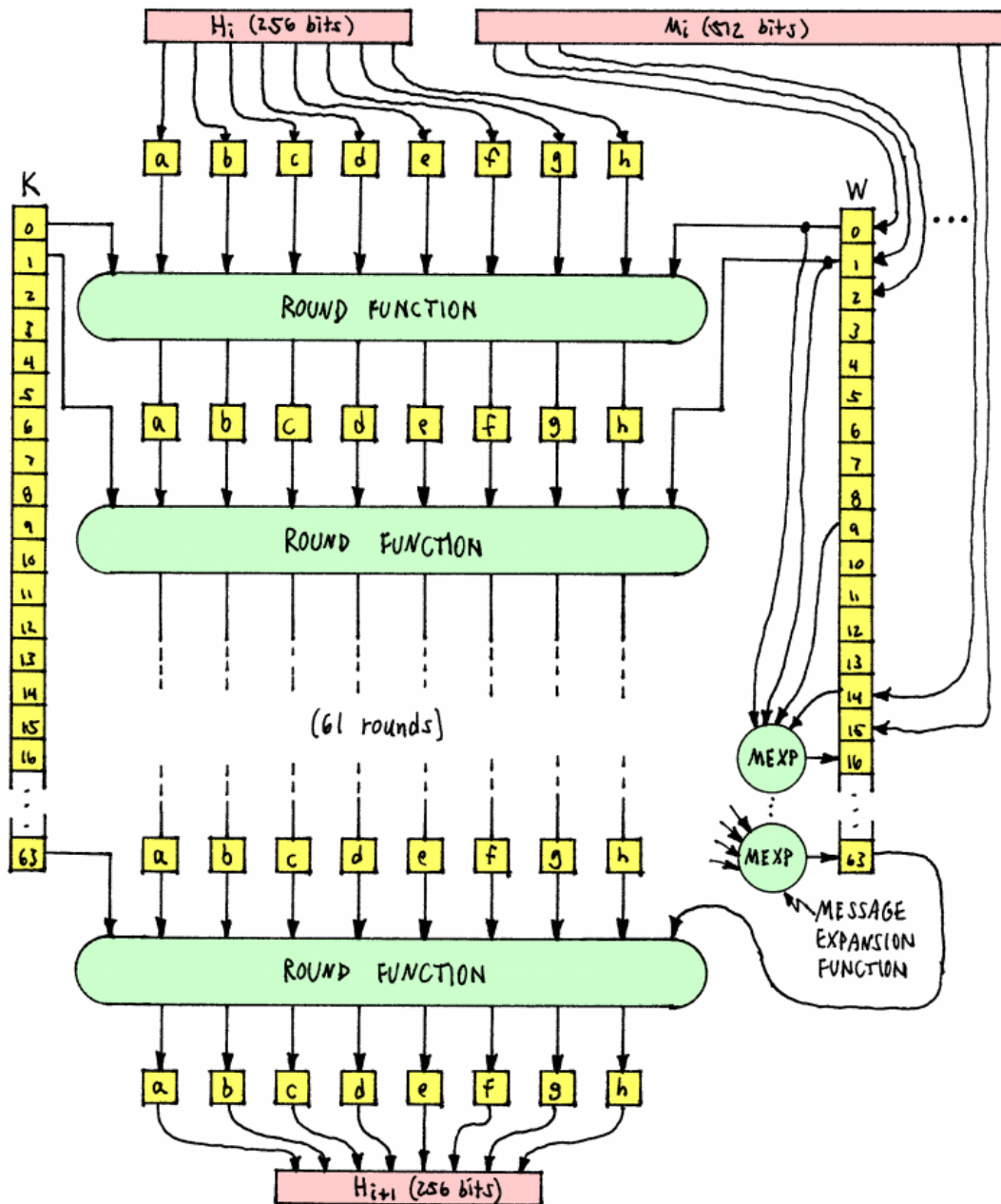


Fig 6.1 SHA-256 COMPRESSION FUNCTION

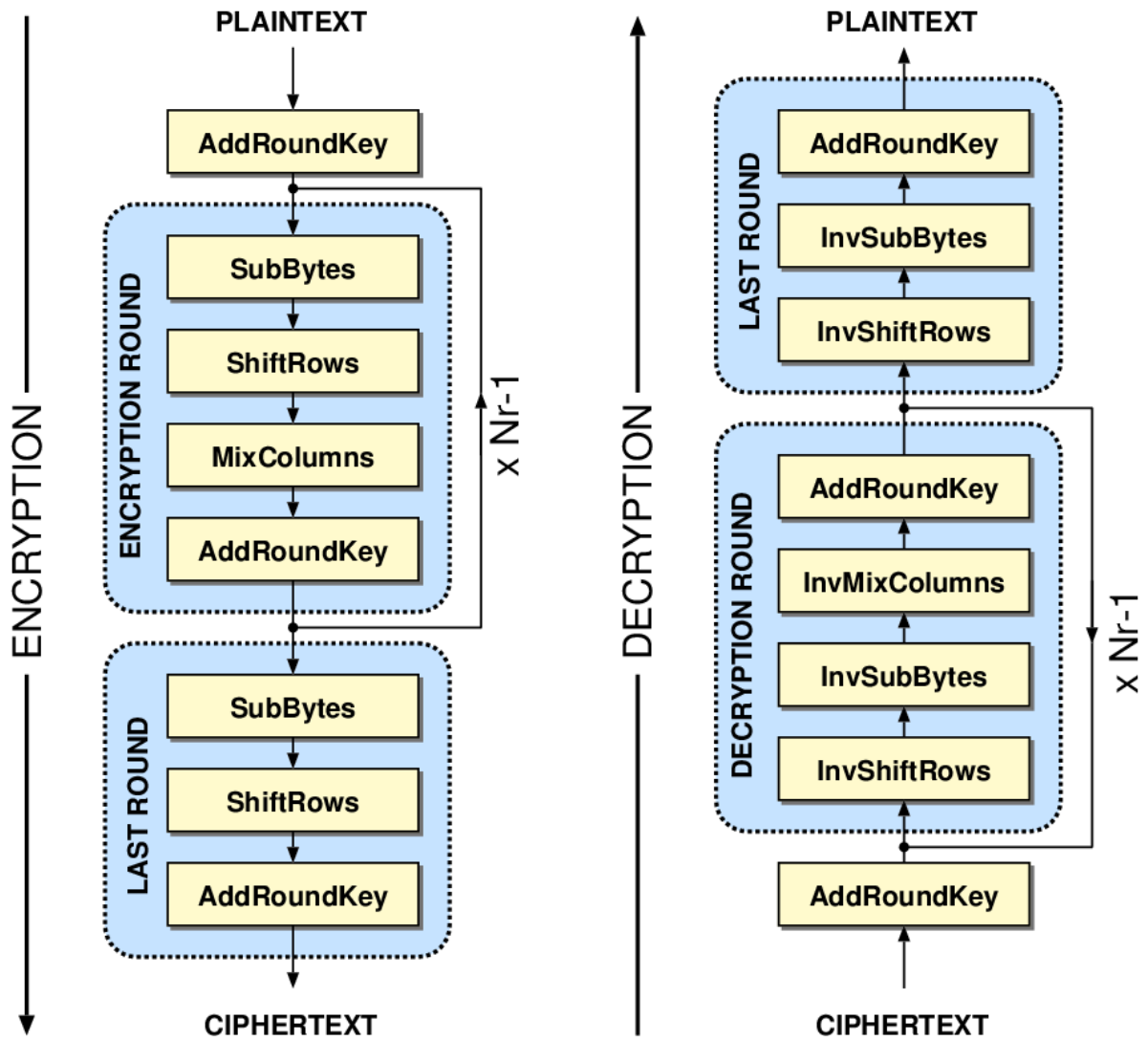
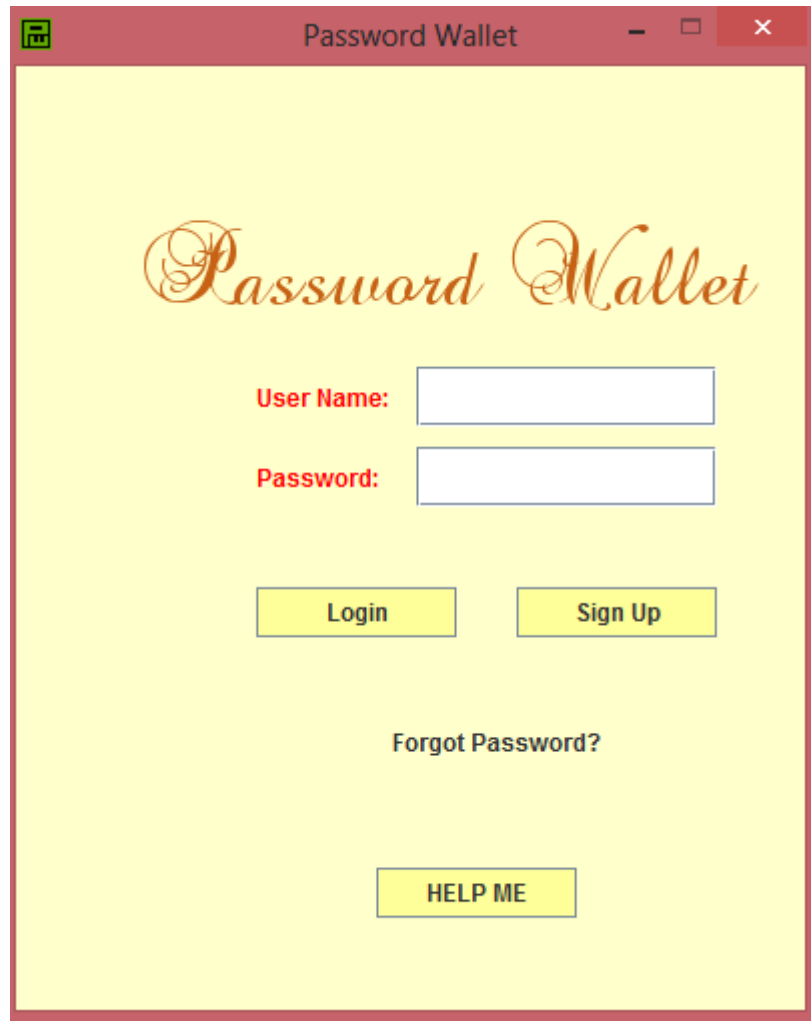


Fig 6.2 Block Diagram of AES Encryption and Decryption

## 6.3 SCREEN SHOTS



Password Wallet

*Password Wallet*

User Name:

Password:

Login Sign Up

Forgot Password?

HELP ME

Fig 6.3 Login screen of Password Wallet

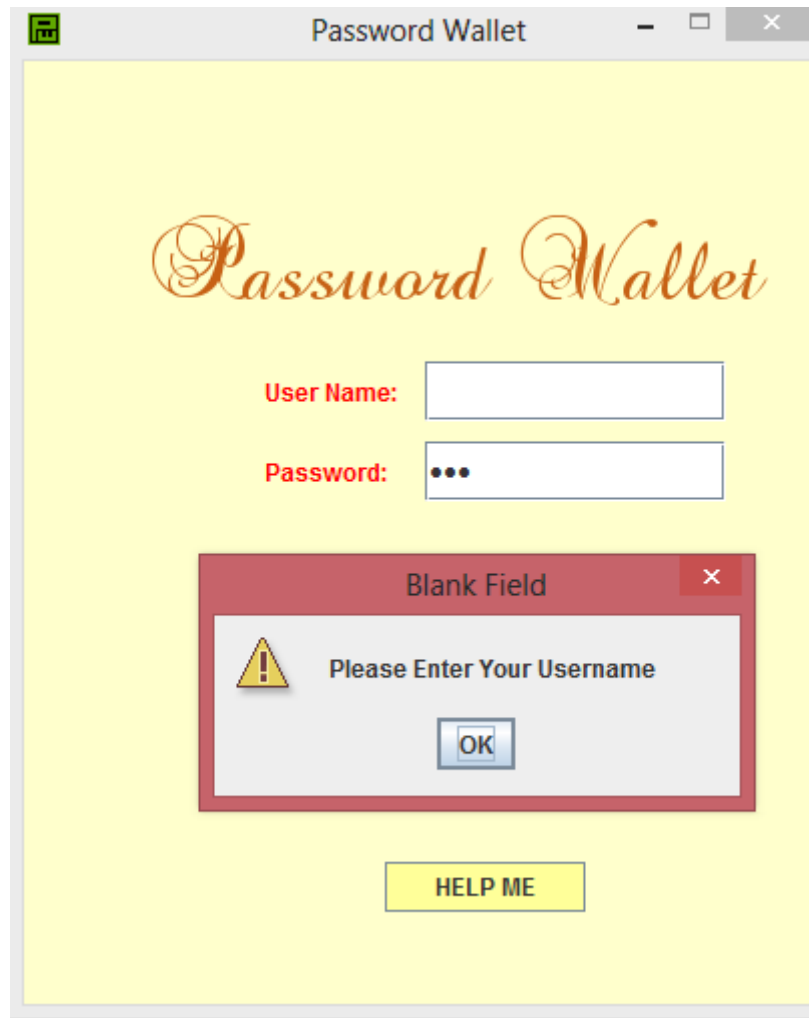


Fig 6.4 Pop up box on not entering username



Fig 6.5 Pop up box if password field is empty



Fig 6.6 Pop up box if invalid username is entered



Fig 6.7 Home Screen of Password Wallet





The image shows a software window titled "Password Wallet" with a red title bar and standard Windows window controls. The main area has a yellow background. At the top, the text "Password Wallet" is written in a large, brown, cursive font. Below this, there are five input fields with red labels: "Enter UserName :" with the text "jill", "Enter Login Password :" with three dots, "Confirm Login Password :" with three dots, "Choose Security Question :" with a dropdown menu showing "What is your first pet ?", and "Enter Security Answer :" with the text "dog". At the bottom, there are two yellow buttons with black text: "CREATE" and "CANCEL".

Enter UserName : jill

Enter Login Password : ...

Confirm Login Password : ...

Choose Security Question : What is your first pet ?

Enter Security Answer : dog

CREATE CANCEL

Fig 6.8 Create Account Frame

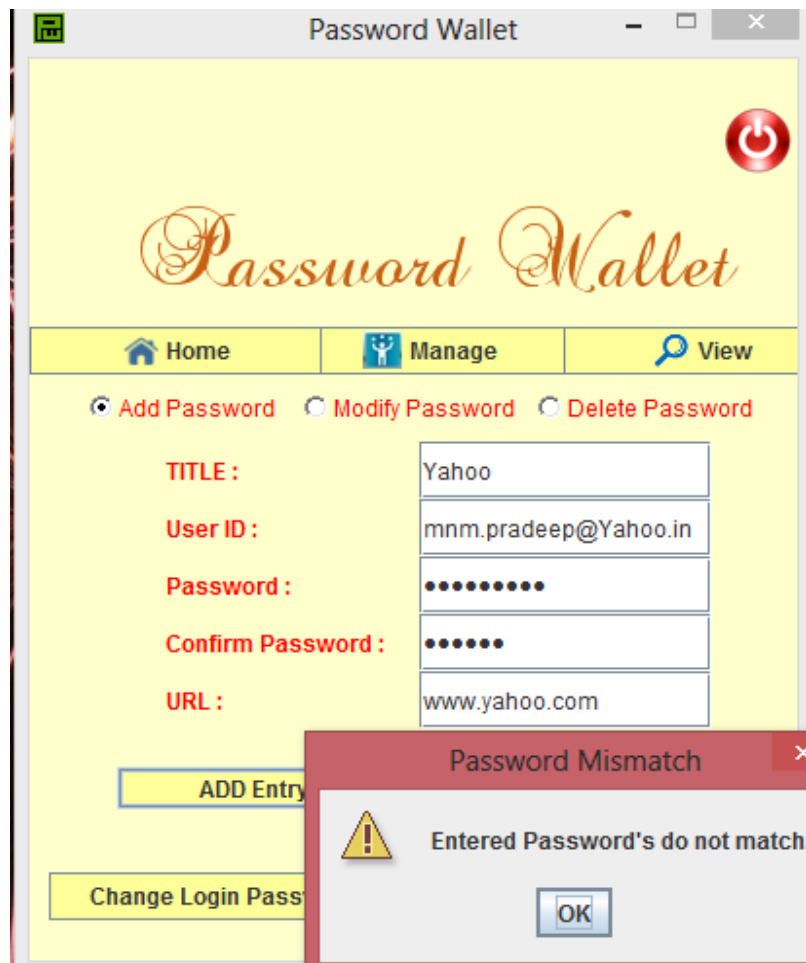


Fig 6.9 If passwords don't match while adding password to Password Wallet

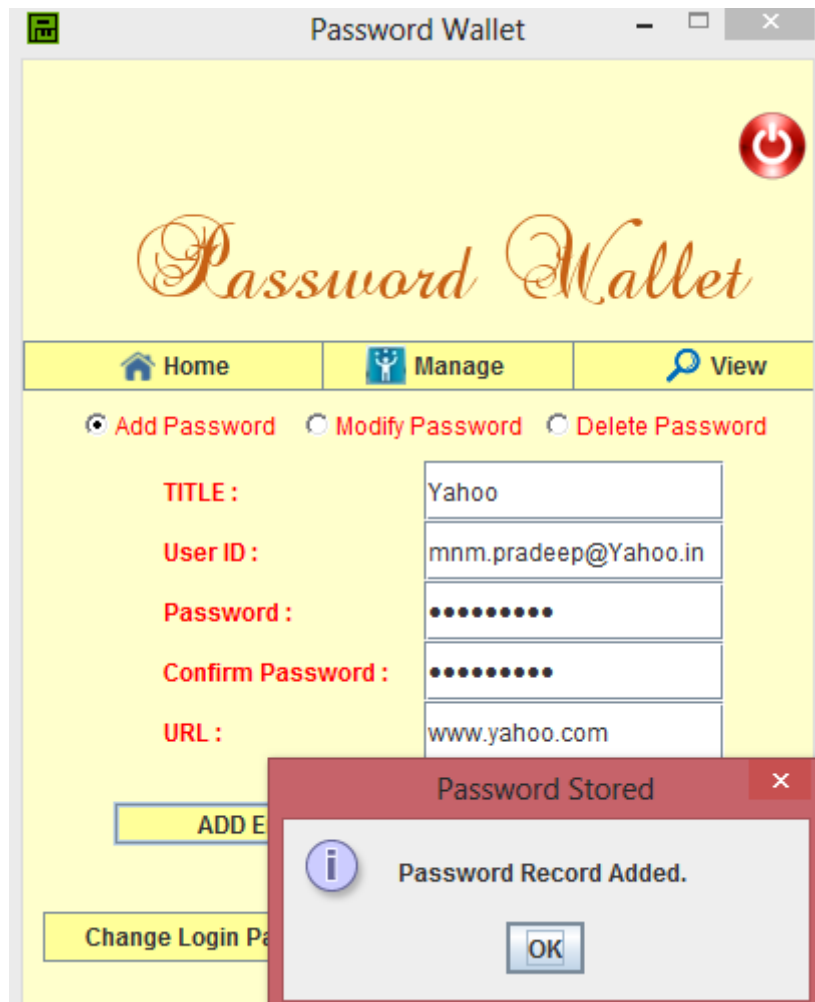
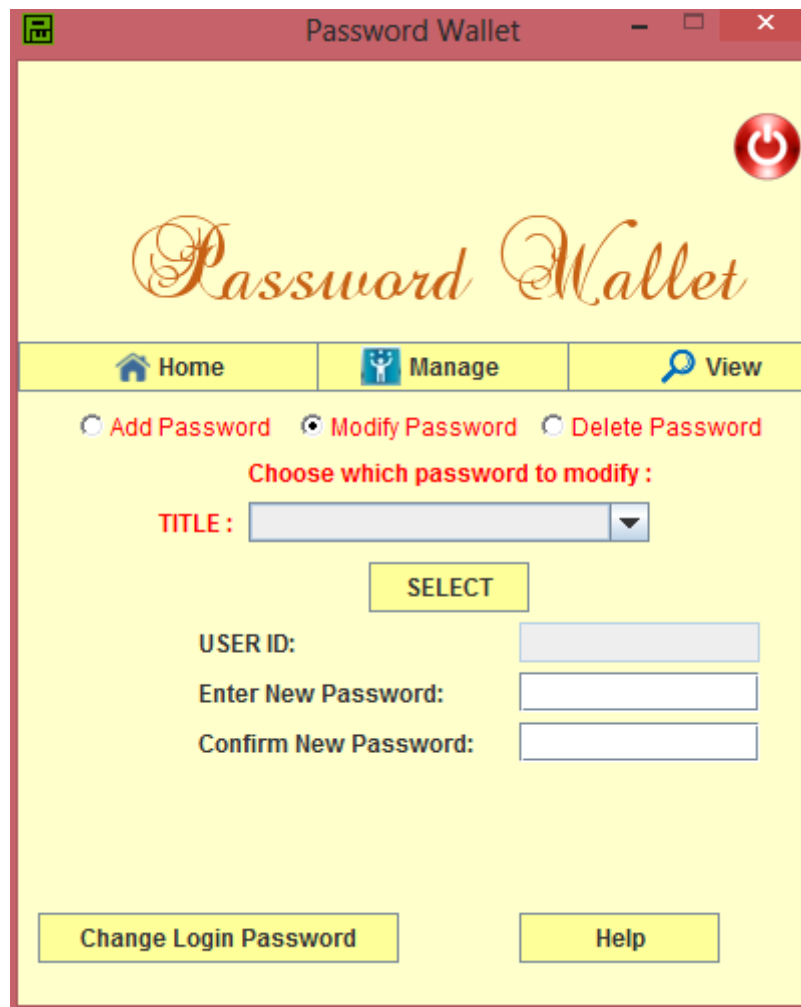


Fig 6.10 Passwords successfully added



The screenshot shows a window titled "Password Wallet" with a red title bar. The window has a yellow background. At the top right, there is a red power button icon. The main title "Password Wallet" is written in a large, elegant, brown script font. Below the title, there is a horizontal menu with three items: "Home" (with a house icon), "Manage" (with a person icon), and "View" (with a magnifying glass icon). Under the "Manage" menu, there are three radio buttons: "Add Password", "Modify Password" (which is selected), and "Delete Password". Below these, the text "Choose which password to modify :" is displayed in red. This is followed by a "TITLE :" label and a text input field with a dropdown arrow. Below the input field is a yellow "SELECT" button. Further down, there are three labels with corresponding input fields: "USER ID:", "Enter New Password:", and "Confirm New Password:". At the bottom of the window, there are two yellow buttons: "Change Login Password" and "Help".

Fig 6.11 Modifying passwords

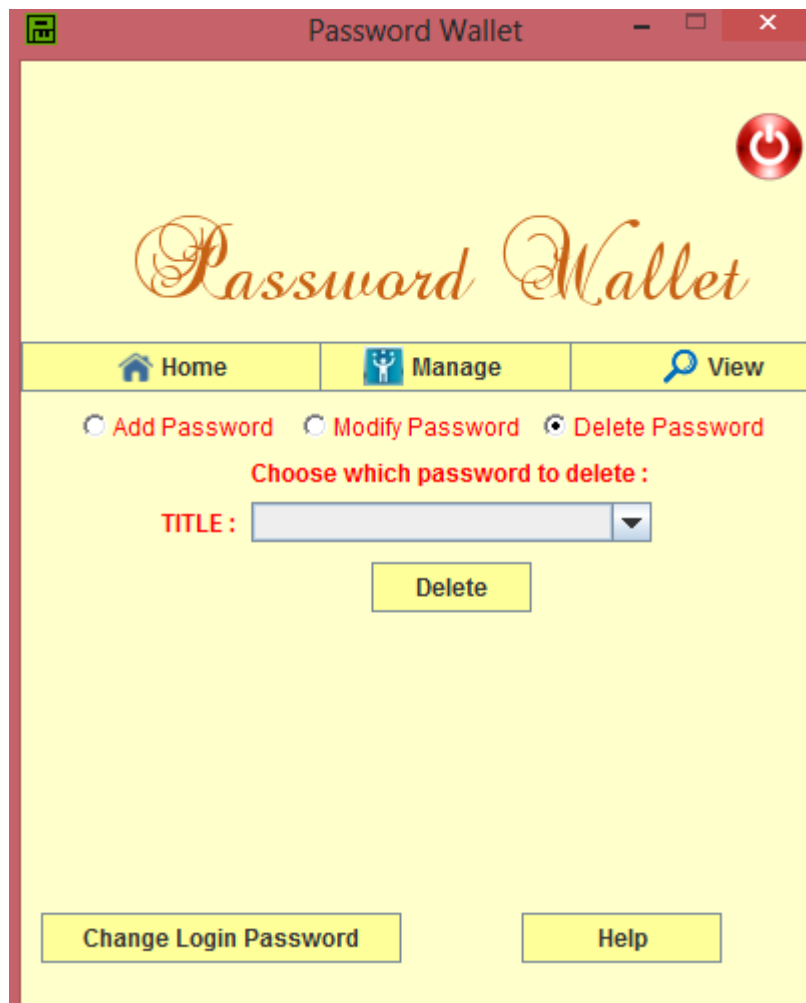
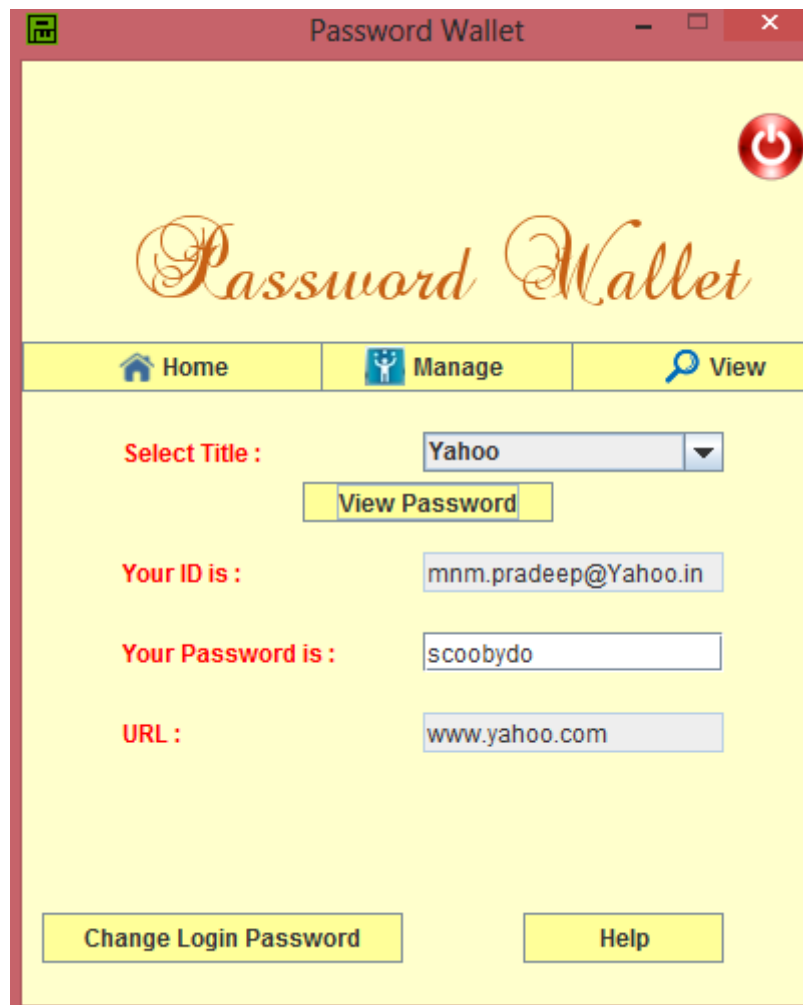


Fig 6.12 Deleting Passwords from Password Wallet



Fig 6.13 Confirmation for deleting password



The screenshot shows a window titled "Password Wallet" with a red title bar. The window has a yellow background and a red power button icon in the top right corner. The main title "Password Wallet" is written in a large, stylized, brown cursive font. Below the title is a navigation bar with three tabs: "Home" (with a house icon), "Manage" (with a person icon), and "View" (with a magnifying glass icon). The "View" tab is currently selected. The main content area has a light yellow background and contains the following fields and buttons:

- Select Title :** A dropdown menu with "Yahoo" selected.
- View Password**: A yellow button.
- Your ID is :** A text input field containing "mnm.pradeep@Yahoo.in".
- Your Password is :** A text input field containing "scoobydo".
- URL :** A text input field containing "www.yahoo.com".
- Change Login Password**: A yellow button.
- Help**: A yellow button.

Fig 6.14 Retrieving Passwords by entering the appropriate title

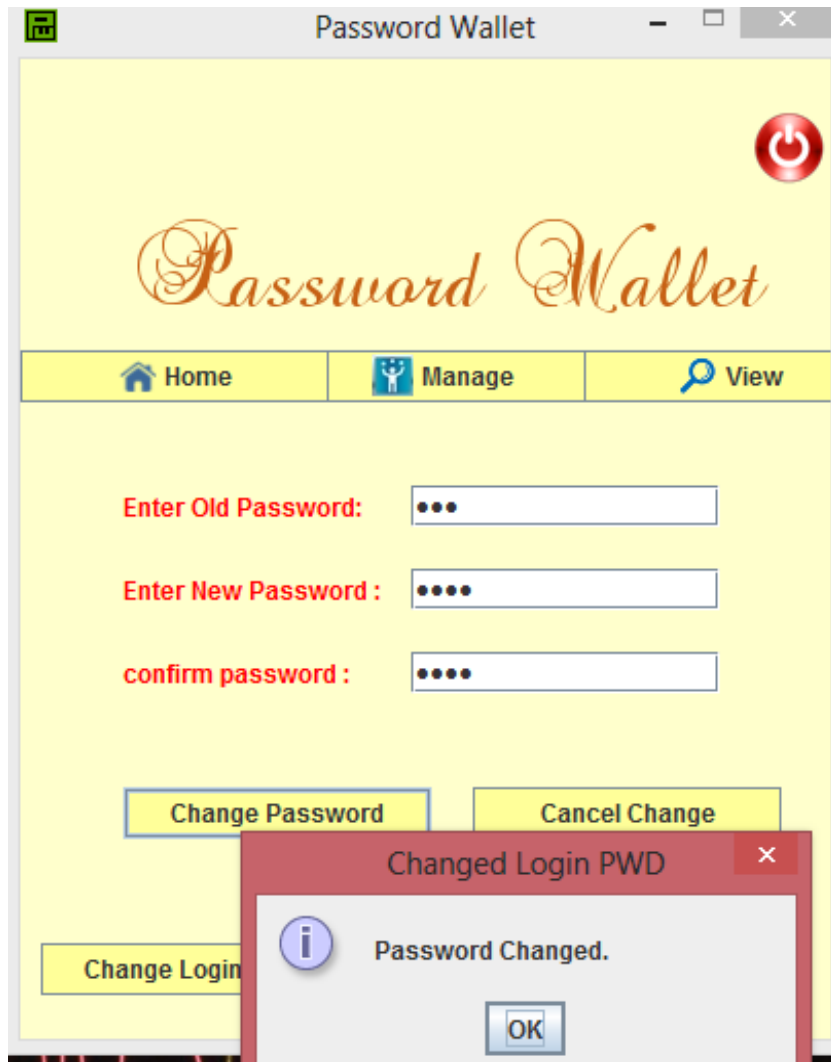
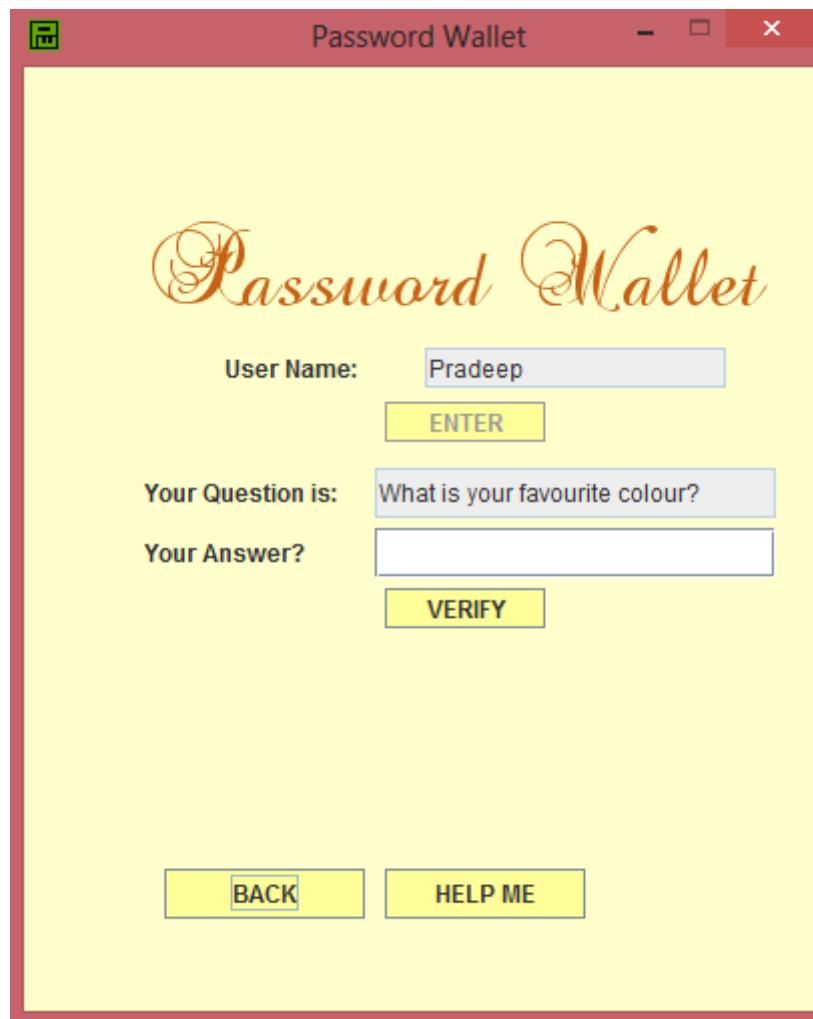


Fig 6.15 Changing Login Password





The image shows a window titled "Password Wallet" with a yellow background and a red border. The title bar includes a green icon, the text "Password Wallet", and standard window controls (minimize, maximize, close). The main content area features the title "Password Wallet" in a large, brown, cursive font. Below the title, there are three input fields and three buttons. The first input field is labeled "User Name:" and contains the text "Pradeep". Below it is a yellow button labeled "ENTER". The second input field is labeled "Your Question is:" and contains the text "What is your favourite colour?". Below it is a yellow button labeled "VERIFY". The third input field is labeled "Your Answer?" and is empty. Below it is a yellow button labeled "BACK". At the bottom of the window, there are two yellow buttons: "BACK" and "HELP ME".

Password Wallet

User Name: Pradeep

ENTER

Your Question is: What is your favourite colour?

Your Answer?

VERIFY

BACK HELP ME

Fig 6.16 Forgot Password Frame

The screenshot shows the Microsoft Access application window with the 'Table Tools' ribbon active. The 'Tables' list on the left includes 'MAIN\_TB', which is currently selected and displayed in the main area. The table is in 'Datasheet View' and contains the following data:

UID	PWD	SEC_QN	SEC_ANS	Add New Field
abc	fed912ec79b3f1c7738e892c0669c8c782bce5fcb743cf	0	abc	
Chandu	56b909135371d6960b71c0048eb36747d69d053ab2ac4	1	chandu	
hello	f6c9737012d25c41df35f6d5d03d8d84912795326ffd5d	1	dog	
hi	fa3ba64f2053ed06fc34ef5d5888983ca6ee22c7bd7d3c	1	joi	
Pradeep	6aa414180d09a8f4917c981826bd83506e1bd926ee996	2	Red	
Raghu	5f82a2d1a8549fe73bed91ffa28d7e4e6f2912cdfa8d20	0	Chintoo	
Raghu_TB	a466bcf149d66ed252c1cb921911963fd9e9fcd808ecc	0	chintoo	
Raghunandan	de678e9e72fb91d022805f15221cefc9812301fbd4315	1	Chicken	
SAIKRISHNA	a0bc3ae5186ab8ee8e4513a5cf15afcc5a809af17a1451	0	chintoo	
Srinidhi	fd2f05a4556e35d408663643b3b0337711a04741c9306c	4	Chicken	

Fig 6.17 MAIN\_TB table in the database

## CHAPTER 7

### TESTING

#### 7.1 PURPOSE

The purpose of testing is to assess product quality. It helps to strengthen and stabilize the architecture early in the development cycle. We can verify through testing, the various interactions, integration of components and the requirements which were implemented. It provides timely feedback to resolve the quality issues, in a timely and cost effective manner. The test workflow involves the following:

- Verifying the interactions of components.
- Verifying the proper integration of components.
- Verifying that all requirements have been implemented correctly.
- Identifying and ensuring that all discovered defects are addressed before the software is deployed.

#### 7.2 DIMENSIONS OF TESTING

To assess product quality, different kinds of tests, each one with a different focus, are needed. These tests can be categorized by several dimensions:

- **Quality dimension:** The major quality characteristic or attribute that is the focus of test.
- **Stage of testing:** The point in the lifecycle at which the test, usually limited to a single quality dimension.
- **Type of testing:** The specific test objective for an individual test, usually limited to a single quality dimension.

## 7.3 TYPES OF TESTING

### 7.3.1 Unit Testing

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

### 7.3.2 Integration Testing

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

### 7.3.3 White Box Testing

The purpose of any security testing method is to ensure the robustness of a system in the face of malicious attacks or regular software failures. White box testing is performed based on the knowledge of *how* the system is implemented. White box testing includes analyzing data flow, control flow, information flow, coding practices, and exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities. White box testing requires access to the source code.

It is a good practice to perform white box testing during the unit testing phase. White box testing requires knowing what makes software secure or insecure, how to think like an attacker, and how to use different testing tools and techniques.

The first step in white box testing is to comprehend and analyze source code, so knowing what makes software secure is a fundamental requirement. Second, to create tests that exploit software, a tester must think like an attacker. Third, to perform testing effectively, testers need to know the different tools and techniques available for white box testing. The three requirements do not work in isolation, but together.

### **7.3.4 Black Box Testing**

It is also known as functional testing- a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications.

The advantages of this type of testing include:

- The test is unbiased because the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

### **7.3.5 System Testing**

System testing validates software once it has been incorporated into a larger system. Software is incorporated with other system elements and a series of system integration and validation tests are conducted. System testing is actually a series of different test whose primary purpose is to fully exercise the computer- based system. Once the system has been developed it has to be tested.

In the present system we have to take care of valid property and assessment numbers i.e. there should not exist any duplicate number in each case. Care should be taken that the appropriate data is retrieved in response to the queries.

### **7.3.6 Validation Testing**

The terms verification and validations are used interchangeably we will describe both these methods. Verification is the process of determining whether or not the products of given phase of software development fulfil the specifications established in the previous

phase. These activities include proving and reviews. Validation is the process of evaluating the software at the end of software development process; we find how well the software satisfies the requirement specifications. The requirement of the software starts with requirement document and requirement specifications without errors and specifying client's requirements correctly. The validation process of evaluating the developed system at the end is to ensure that it must satisfy all the necessary requirement specification. Requirement verification also checks the factors as completeness, consistency and testability of the requirements. As we all know that testing plays a crucial role in evaluation of the system. That is in order to know whether the system working properly or not. In other words we can say that in order to know whether the system which we have developed will give the expected output or not can be known by doing the testing. Testing phase comes after coding phase. Usually organizations or the software developing companies use different types of testing strategies in order to evaluate the performance of a system. Also it gives the output which provides clear information regarding the project or system, whether the project which we have developed will going to give the expected output or not, that is whether the system fails or succeed in the market. We have many types of testing such as unit testing, integration testing, system testing, and black box testing, white box testing and regression analysis testing and so on. In our project secure cryptographic messaging we are using unit testing, integration testing, and system testing. Unit testing is the one in which each entity or objects in the module will be tested.

That's once unit testing is done with all modules, than integration testing will be done, on the every module or on group of two or three modules. Finally system testing will be done , in which all the modules of a system will be tested at once , there by getting the overall performance of a system that means we can conclude the result on the entire system whether our system is working as per our requirements or as per our expectations or not. The advantage of developing or testing modules wise is that, we can reduce the effort, cost and time.

## 7.4 TEST CASES

Test ID	Function	Expected Result	Observed Result	Status	Reference
1.	Checking if username field is entered or not.	Error-message requesting the user to enter the user name should pop up.	If username field is empty, error message pops up.	Pass	Refer to Fig 6.4
2.	Checking if password field is entered or not.	Error-message requesting the user to enter the password should pop up.	If password field is empty, error message pops up.	Pass	Refer to Fig 6.5
3.	Authentication	If given credentials are valid, then redirect to the home-page.	Redirected to the respective page.	Pass	Refer to Fig 6.6
4.	Retrieving passwords on giving the accurate titles.	If the title given by user matches the title in database then retrieve the password.	The password is retrieved from the database.	Pass	Refer to Fig 6.14
5.	Checking if username is valid or not.	If entered username does not match the username in database then error message pops up.	If username is not valid, an error message pops up.	Pass	Refer to Fig 6.7
6.	Checking if entered password and confirmation passwords match.	If entered password does not match confirmation password then an error message pops up.	If passwords don't match then an error message pops up.	Pass	Refer to Fig 6.10

## **CHAPTER 8**

### **CONCLUSION AND FUTURE ENHANCEMENT**

If you are using ‘123456’, ‘abc123’, or ‘password’ you are not alone. In fact these are some of the most commonly used passwords. Using one of these is like leaving the key to your front door under the welcome mat! The problem is that difficult to crack passwords are also difficult to remember. A password such as “U#8?\*e1v” while strong, is not easy to remember, let alone type into a password field that only displays dots. The issue is further compounded when you consider that for maximum security your passwords need to be unique for every login. On some systems you may also be expected to change your password from time to time.

Thus, this project aims at creating an application which can manage all the passwords for a user, without compromising on security. Users do not have to worry about remembering dozens of passwords as every password is locked up safely for him which can be unlocked only by a key (title). The AES encryption/decryption and SHA algorithms used make sure that the system remains secure against brute-force attack or any kind of security attack. The graphical user interface facilitates an easy interaction for the users with the application.

This application can further be enhanced on the security front by encrypting the entire database. In such a case all the database fields including the username will be protected from the intruders. Another major improvement in this application can be the inclusion of a password generator which can automatically generate a strong password for the user. This can save the user’s time and also increase the strength of the passwords in the database.



## REFERENCES

- [1] Cryptography and Network Security, B.A. Forouzan, D.Mukhopadhyay, 2<sup>nd</sup> Edition
- [2] Network Security Essentials( Applications and Standards) by William Stallings
- [3] Core Java by Dr R. Nageswara Rao
- [4] Software Engineering: A Practitioner's Approach, Roger S Pressman, 6<sup>th</sup> edition
- [5] Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide
- [6] [http://en.wikipedia.org/wiki/Password\\_manager](http://en.wikipedia.org/wiki/Password_manager)
- [7] <http://bradleyandcompany.com/password-managers/>
- [8] <http://stackoverflow.com/questions/18847580/aes128-vs-aes256-using-bruteforce>