

Term Project CS 151:  
Functional Spec

Object-Oriented Design  
**Prof: Ahmad Yazdankhah**

<b>Team Number</b>	Team #4	<b>Section</b>	02
<b>Team Members</b>	<ol style="list-style-type: none"><li>1. Mohit Sonwane</li><li>2. Patrick Chau</li><li>3. Oliver Mancebo</li><li>4. Artharv Mangal</li></ol>		
<b>Software Name</b>	Keyware		

## 1. Problem Statement

Computers and the Internet have become essential tools for business, education, entertainment, and so many other facets of daily life. The multitude of accounts required to utilize online services or computer applications can cause users to choose easily memorable passwords, or simply reuse old passwords. These passwords can also be stored in an unsafe manner, introducing many vulnerabilities into a user's personal information. Strong cyber security is paramount in protecting these passwords, login credentials, and the sensitive information locked behind them. As data breaches and online security compromises become more common in the digital age, the risk of sensitive information being leaked requires better solutions for protecting user data. Password managers can be useful in creating more secure passwords and safely storing them, although current password manager solutions are still at risk from using weak methods to encrypt the user data, and the developers having access to a user's information. An improved password manager should be developed that is easy to use, supports the creation of secure random passwords, and utilizes strong encryption of passwords.

## 2. Functional Requirements

1. The application will support the creation of multiple user accounts, each with access to their own individual database of stored passwords after logging into the application.
2. Each user can create their own account and change their account password (by using a security question that is set up during the signup process).
3. The user will be able to store an existing password for a website/application, along with the username, email, date of storing, and expiration date of that password.

4. The user will be able to generate a strong random password for a website/application, along with the username, email, date of creation, and expiration date of that password.
5. The application will provide an extra level of security by encrypting passwords before storing in the application's database, and decrypting passwords before the user retrieves the password.
6. The user will be able to modify the information associated with a password (including the password itself, username, email, and application/website name) and copy the password to their clipboard.
7. After the user logs in, the application will search through the user's passwords and notify the user of any expired passwords.
8. The user can quickly find a password by searching with an associated username or website/application name.

### **3. References**

**N/A**

## 4. Use Cases

Use Case Name		Sign Up
Goal		
This use case describes how a user creates a new account for the password manager.		
Participating Actors		
User, system		
Glossary		
N/A		
Primary Flow of Events		
Trigger		
User clicks on the “create account” button.		
Steps	Action	System Response
1	User selects create account button on login page	System redirects the user to create an account page. System asks for user's email address, username, choice of security question/answer, and password (password requested twice for verification)
2	User inputs new username and password and email address. User chooses the security question and provides the answer to it. User clicks create account button	System checks that username and email are not already associated with an account. System checks that passwords match. System encrypts the login password, stores it, and creates the account. System redirects the user back to the login page.
Alternate Flow of Events		
Alternate Trigger		
User exits out of the create account page		
Steps	Action	System Response
1	User closes the create account page, or clicks “go back” button, before creating the account	System does not create an account or save any entered information. System takes the user back to the login page if “go back” is selected
Alternate Trigger		
User does not enter username, password, or email address		

Steps	Action	System Response
1	User clicks create account button but information is missing	System notifies the user of missing information and prompts the user to enter it. System does not create account until all information is entered
<b>Alternate Trigger</b>		
Username or email provided is already associated with an account		
Steps	Action	System Response
1	User clicks create account button but the username or email is already registered to an account	System notifies the user that email or username is already associated with an account. System does not create the account until an unregistered username or email is entered.
<b>Alternate Trigger</b>		
The passwords the user enters do not match		
Steps	Action	System Response
1	User enters passwords which do not match	System notifies the user that the passwords do not match. System does not create the account until both passwords match.

<b>Use Case Name</b>		Log In
<b>Goal</b>		
User enters the home page of password manager after they enter their login credentials		
<b>Participating Actors</b>		
User, system		
<b>Glossary</b>		
N/A		
<b>Primary Flow of Events</b>		
<b>Trigger</b>		
User opens password manager application		
Steps	Action	System Response
1	User opens app to the login screen	System asks for username and password

2	User enters the credentials and clicks login button	System checks if the user has an account with username and password entered. If the account exists, the system directs the user to the home page. System searches the database for any expired password and notifies the user after login.
<b>Alternate Flow of Events</b>		
<b>Alternate Trigger</b>		
<b>User entered wrong credentials</b>		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	Users spell the username/password wrong or miss a character. Or the user enters wrong credentials.	The system displays “invalid username/password”. System does not log in.
<b>Alternate Trigger</b>		
<b>User enters wrong login credentials repeatedly</b>		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User keeps entering wrong password for a correct username	Prohibits the user from entering password temporarily for the same username and sends an automated email to the email associated with that username about multiple login attempts.
<b>Alternate Trigger</b>		
<b>No user passwords are expired</b>		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User logs into their account and no stored passwords are expired	After login, system will not show the expiration notification to the user

<b>Use Case Name</b>	<b>User Login Password Reset</b>
<b>Goal</b>	
This use case describes how the user will be able to reset their login password for this application.	
<b>Participating Actors</b>	
User, system	
<b>Glossary</b>	
N/A	
<b>Primary Flow of Events</b>	

<b>Trigger</b>		
User runs the application and clicks on forgot password button on login page		
Steps	Action	System Response
1	User presses on “Forgot Password” button on login page	System redirects the user to the “Reset Password” screen. System asks the user to enter a username and new password. System asks users to enter answers for security questions.
2	User enters username and new user login password into “Username”, “New Password” and “Confirm Password” fields, then answers security question specified by entering answer in “Security Question Answer” field. Then press Enter key or “Reset Password” button.	System verifies username and security question answers. System checks that the new password is different from the old password. System replaces user’s old password with new, specified password in system. System encrypts this password before storing it. Redirects to log in screen. System shows black message at top: “Password reset successful!”
<b>Alternate Flow of Events</b>		
<b>Alternate Trigger</b>		
The user runs the application and clicks on the “Reset Password”. User clicks on the “go back” button or exits the page.		
Steps	Action	System Response
1	The user clicks on the “go back” button on the “Reset Password” screen.	System redirects user to Login screen
<b>Alternate Trigger</b>		
The user enters incorrect Username/security question answers.		
Steps	Action	System Response
1	The user presses Enter key or “Reset Password” button.	System shows a red message under the “reset password” field: “Your username and/or security question answer was incorrect. Please try again.”  The system removes all text from “Username”, “New Password”, “Confirm Password”, and “Security Question Answer” fields.  The system stays on the “Reset Password” screen.
<b>Alternate Trigger</b>		
The user enters new passwords which do not match		
Steps	Action	System Response

1	User enters passwords which do not match and clicks on the reset password button.	System notifies the user that the passwords do not match. System stays on the reset password screen and does not update the password until both passwords match.
<b>Alternate Trigger</b>		
The user enters new passwords which are the same as old password		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User enters passwords which are not different from the old password, and clicks on the reset password button.	System notifies the user that the passwords are the same as the old password. System stays on the reset password screen and does not update the password until new passwords are entered.

<b>Use Case Name</b>		<b>Generate Password</b>
<b>Goal</b>		
This use case describes how a user generates a new random password in the password manager.		
<b>Participating Actors</b>		
User, system		
<b>Glossary</b>		
N/A		
<b>Primary Flow of Events</b>		
<b>Trigger</b>		
User clicks on the generated password.		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User selects the generate password option.	System generates a random password. System asks for username, website/application name, and email associated with the new password.
2	User inputs the username, website/application name, and email associated with the new password. User selects save password.	System checks that username/email and website/application are not already associated with an existing saved password. System encrypts the password and stores it. System stores the date of creation and creates an expiration date for the password. System exits the password generator page.

Alternate Flow of Events		
Alternate Trigger		
User exits out of the generate password page		
Steps	Action	System Response
1	User closes the generate password page before saving	System does not create password or save any entered information
Alternate Trigger		
User does not enter username, website/application name, or email address		
Steps	Action	System Response
1	User clicks save password button but information is missing	System notifies the user of missing information and prompts the user to enter it. System does not save the password until all information is entered
Alternate Trigger		
Username/email and application/website name provided is already associated with a saved password		
Steps	Action	System Response
1	User clicks save password button but the username/email and application/website is already associated with an existing password	System notifies the user that email/username and application/website is already associated with an existing password. System asks the user if they would like to overwrite the existing password
2	User selects to overwrite the existing password	System updates the password associated with the provided email/username and application/website. System encrypts the password and stores it. System stores the new date of creation and creates an expiration date for the password. System exits the password generator page.
Alternate Trigger		
User selects to not overwrite the existing password in step 2 of the previous alternate flow		
Steps	Action	System Response
1	User selects to not overwrite the existing password	System exits the generated password page. System does not save the generated password.

Use Case Name	Store existing user password
---------------	------------------------------



<b>Goal</b>		
This use case describes how a user stores an existing password in the password manager application		
<b>Participating Actors</b>		
User, system		
<b>Glossary</b>		
N/A		
<b>Primary Flow of Events</b>		
<b>Trigger</b>		
User clicks on add password button in the main page of password manager app		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User clicks on add password button	System asks for username, website/application name, password (twice for verification), and email associated with the password.
2	User inputs the username, website/application name, password, and email associated with the password. User clicks on the save button.	System encrypts the password and stores it and displays the message: "Password saved successfully". System also creates the creation date and expiration date of the stored password. System returns to home page.
<b>Alternate Flow of Events</b>		
<b>Alternate Trigger</b>		
User closes add password page or clicks on cancel button		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User exits out of add password page	System does not save the password and redirects user to login screen
<b>Alternate Trigger</b>		
Username, website/application name, password, or email is not entered		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User clicks on the save button within the add password page.	System notifies user of missing information. The system stays on the reset password screen until all information is entered.
<b>Alternate Trigger</b>		
Passwords entered do not match		

Steps	Action	System Response
1	User enters passwords which do not match and clicks on the save password button.	System notifies the user that the passwords do not match. System stays on the add password screen and does not save the password until both passwords match.

Use Case Name		Modify Existing Password in Manager
Goal		
This use case describes how the user will be able to modify an existing password within the system.		
Participating Actors		
User, system		
Glossary		
N/A		
Primary Flow of Events		
Trigger		
The user pressed on the Modify Password button next to any password in the main page of the application.		
Steps	Action	System Response
1	User presses on the edit password icon next to any password	System creates a field right below the password with a field caption "New Password," "Confirm Password" and two buttons "Change" and "Cancel".
2	User enters a new password into "New Password" and "confirm password" fields and clicks on the "Change" button.	System verifies the new password is different from the old password and that new passwords match. System replaces the old password with the new password (and encrypts it) within the password manager/database.
Alternate Flow of Events		
Alternate Trigger		
The user clicks on the edit password icon and then the "cancel" button		
Steps	Action	System Response
1	The user clicks on the "cancel" button.	System removes the "New Password" fields and the two buttons, "Change" and "Cancel".

Alternate Trigger		
The user does not enter anything into the “New Password” or “Confirm Password” field and clicks the “change” button		
Steps	Action	System Response
1	The user clicks on the “Change” button.	System recognizes the field is empty. The system shows a red message right below the field saying: “Empty Field!”. Nothing else happens.
Alternate Trigger		
The user enters new passwords which do not match		
Steps	Action	System Response
1	User enters passwords into “New Password” and “Confirm Password” fields which do not match and clicks on “change” button.	System notifies the user that the passwords do not match. System stays on the “modify password” screen and does not update the password until both passwords match.
Alternate Trigger		
The user enters new passwords which are the same as old password		
Steps	Action	System Response
1	User enters passwords which are not different from the old password, and clicks on the “change” button.	System notifies the user that the passwords are the same as the old password. System stays on the “modify password” screen and does not update the password until new passwords are entered.

Use Case Name	Search for password
Goal	
This use case describes how a user searches for a stored password in the password manager application.	
Participating Actors	
User, system	
Glossary	
N/A	
Primary Flow of Events	
Trigger	

User enters information into the search password field in the main page of the password manager application, and presses enter.		
Steps	Action	System Response
1	User clicks on the search password field, and enters username or website/application name. User hits enter	System searches for passwords associated with the username or website/application name. System displays list of passwords (in the format of *****) associated with the username or website/application name.
2	User clicks on show password icon for the password they want to display	System decrypts the password and displays it.
Alternate Flow of Events		
Alternate Trigger		
User enters information into the search password field but hits escape or does not press enter.		
Steps	Action	System Response
1	User enters information but presses escape or does not hit enter.	System does not search the password.
Alternate Trigger		
Username or website/application name is not entered		
Steps	Action	System Response
1	User clicks on search password button with no entered information	System notifies the user of missing information. The system does not search until information is entered.
Alternate Trigger		
Username or website/application name is not associated with any stored passwords		
Steps	Action	System Response
1	User enters the username or website/application name and hits enter in the search password field.	System notifies the user that no existing password is stored associated with the username or website/application. System prompts users to enter a new username or website/application name.

Use Case Name	Copying the password
Goal	
To allow users to copy the password.	
Participating Actors	
User, system	

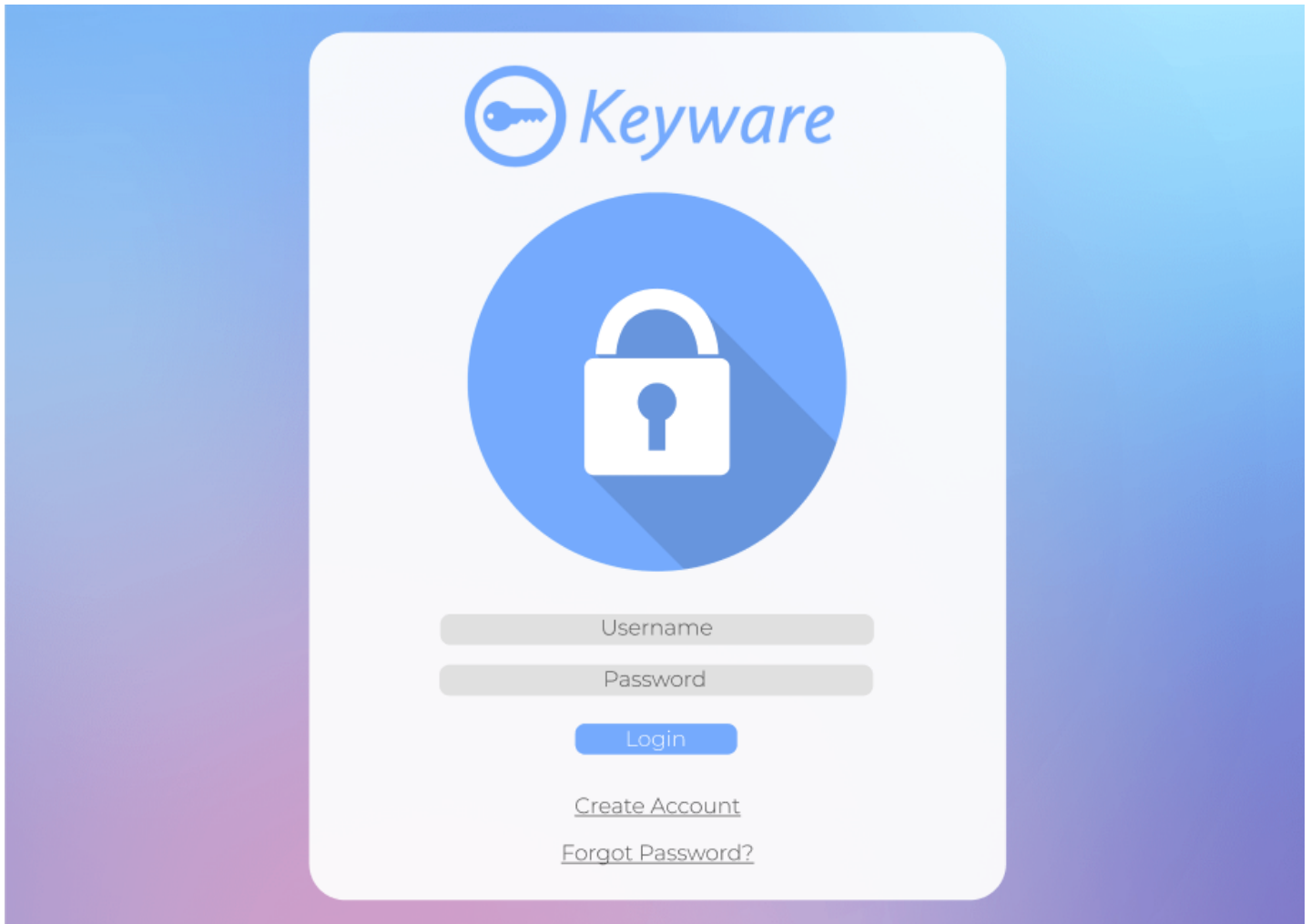
<b>Glossary</b>		
N/A		
<b>Primary Flow of Events</b>		
<b>Trigger</b>		
User clicks on “copy password” icon next to a password		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User hits copy password icon next to a password	The system decrypts the password and copies the passwords
2	User hits Ctrl+v or paste	The system pastes the password.
<b>Alternate Flow of Events</b>		
<b>Alternate Trigger</b>		
User does not click copy button		
<b>Steps</b>	<b>Action</b>	<b>System Response</b>
1	User doesn't want to copy password so ignores copy button	System does nothing

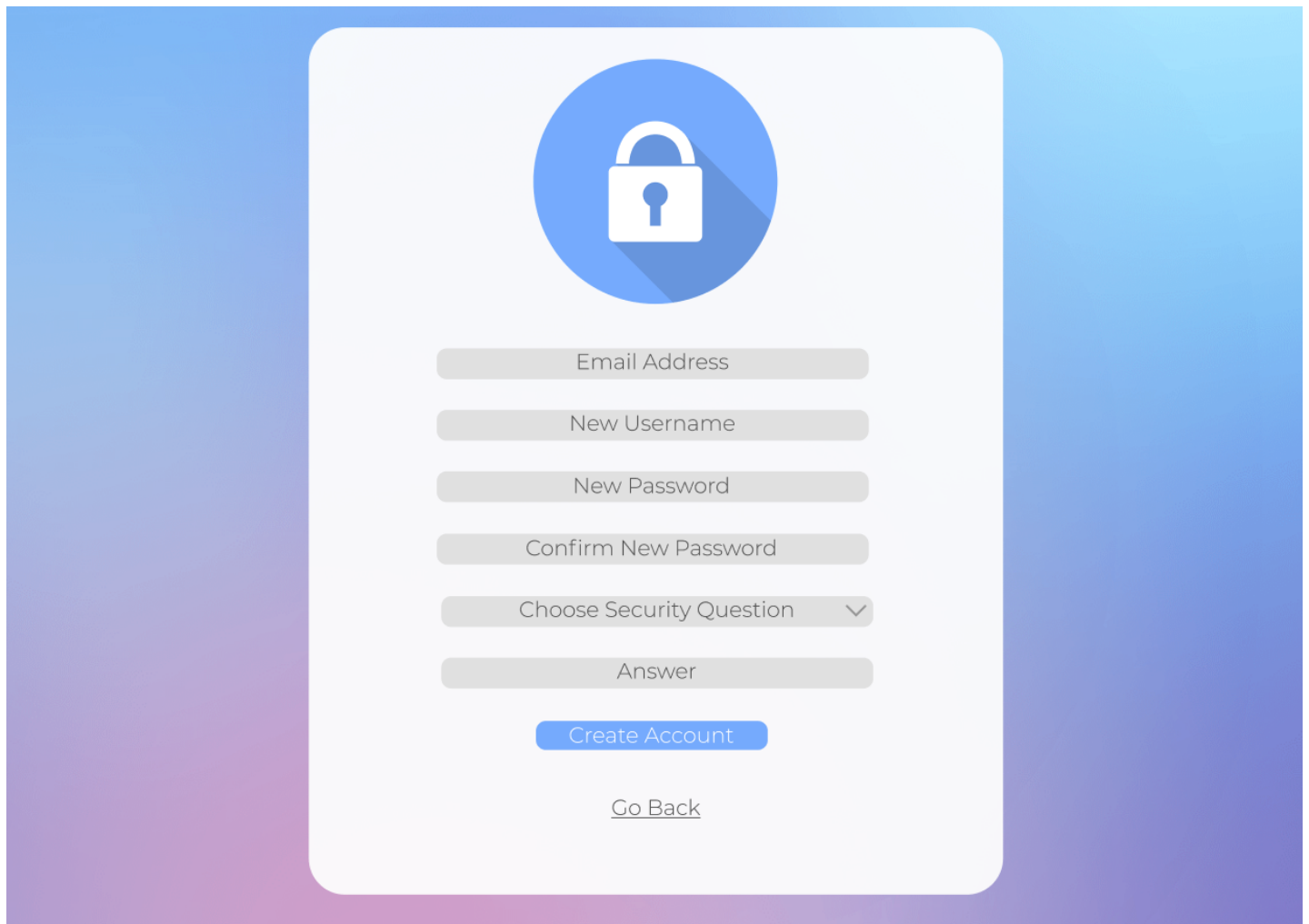
Use Case Name		Log Out
Goal		
User clicks on the logout button in the home page of the password manager		
Participating Actors		
User, system		
Glossary		
N/A		
Primary Flow of Events		
Trigger		
User clicks the log out button		
Steps	Action	System Response
1	User clicks log out	System exits the password manager home page, and returns the user back to the login screen

Alternate Flow of Events		
Alternate Trigger		
User does not click the logout button		
Steps	Action	System Response
1	User does not choose to logout	System keeps user logged into the main password manager home page


## 5. Mockups

### a. Login



**b. Sign Up**

A sign-up form is centered on a light gray rounded rectangle, which is set against a blue-to-purple gradient background. At the top of the form is a circular icon containing a white padlock. Below the icon are seven input fields: 'Email Address', 'New Username', 'New Password', 'Confirm New Password', 'Choose Security Question' (with a dropdown arrow), and 'Answer'. A blue 'Create Account' button is positioned below the 'Answer' field, and a 'Go Back' link is at the bottom of the form.



Email Address

New Username

New Password

Confirm New Password

Choose Security Question ▼

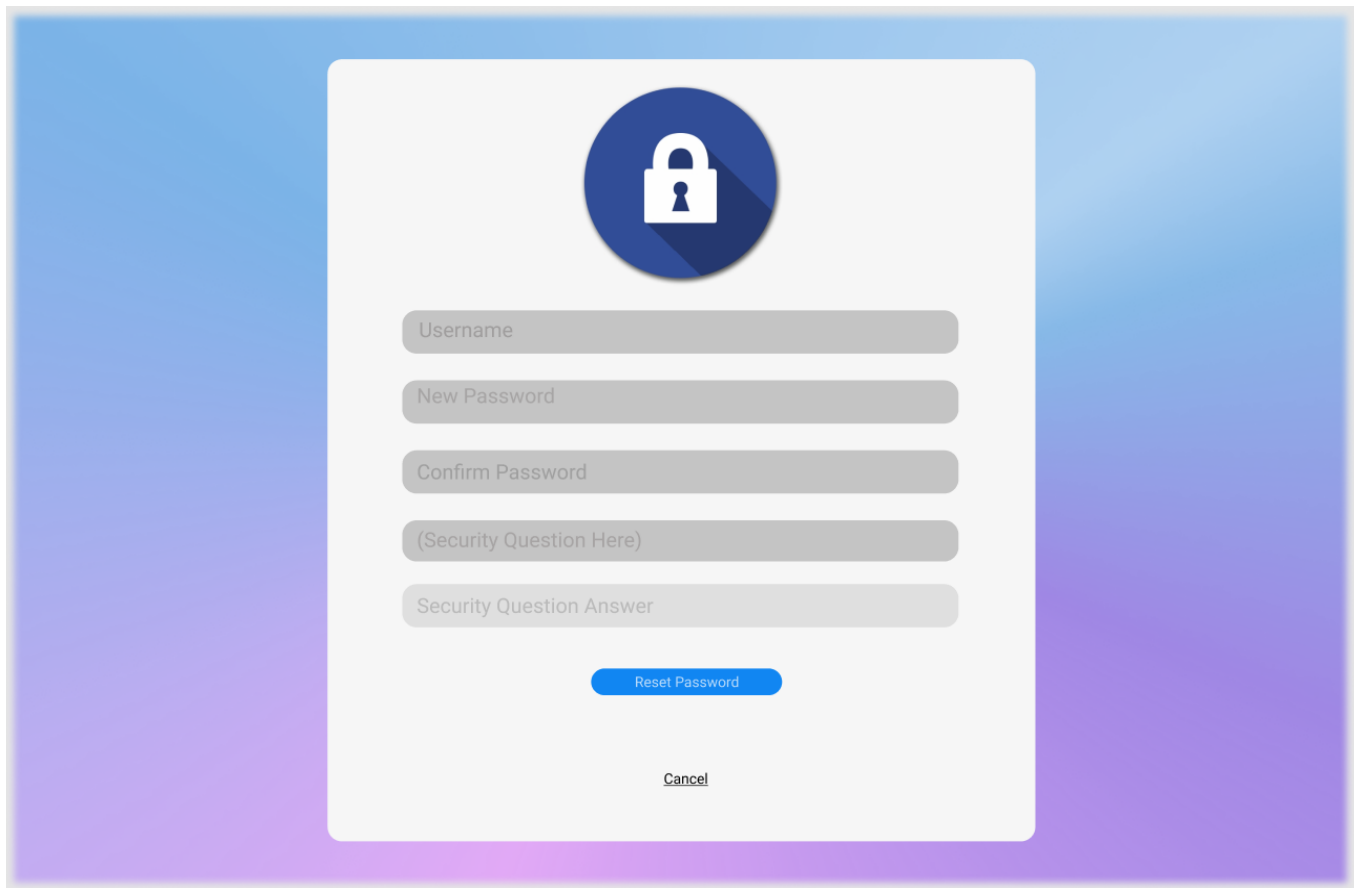
Answer

Create Account


[Go Back](#)



### c. Reset Password



The image shows a 'Reset Password' form centered on a light blue background with a purple gradient at the bottom. The form is a white rounded rectangle. At the top center of the form is a dark blue circular icon containing a white padlock. Below the icon are five text input fields, each with a light gray placeholder text: 'Username', 'New Password', 'Confirm Password', '(Security Question Here)', and 'Security Question Answer'. Below these fields is a blue button with the text 'Reset Password'. At the bottom center of the form is a link labeled 'Cancel'.



Username

New Password

Confirm Password

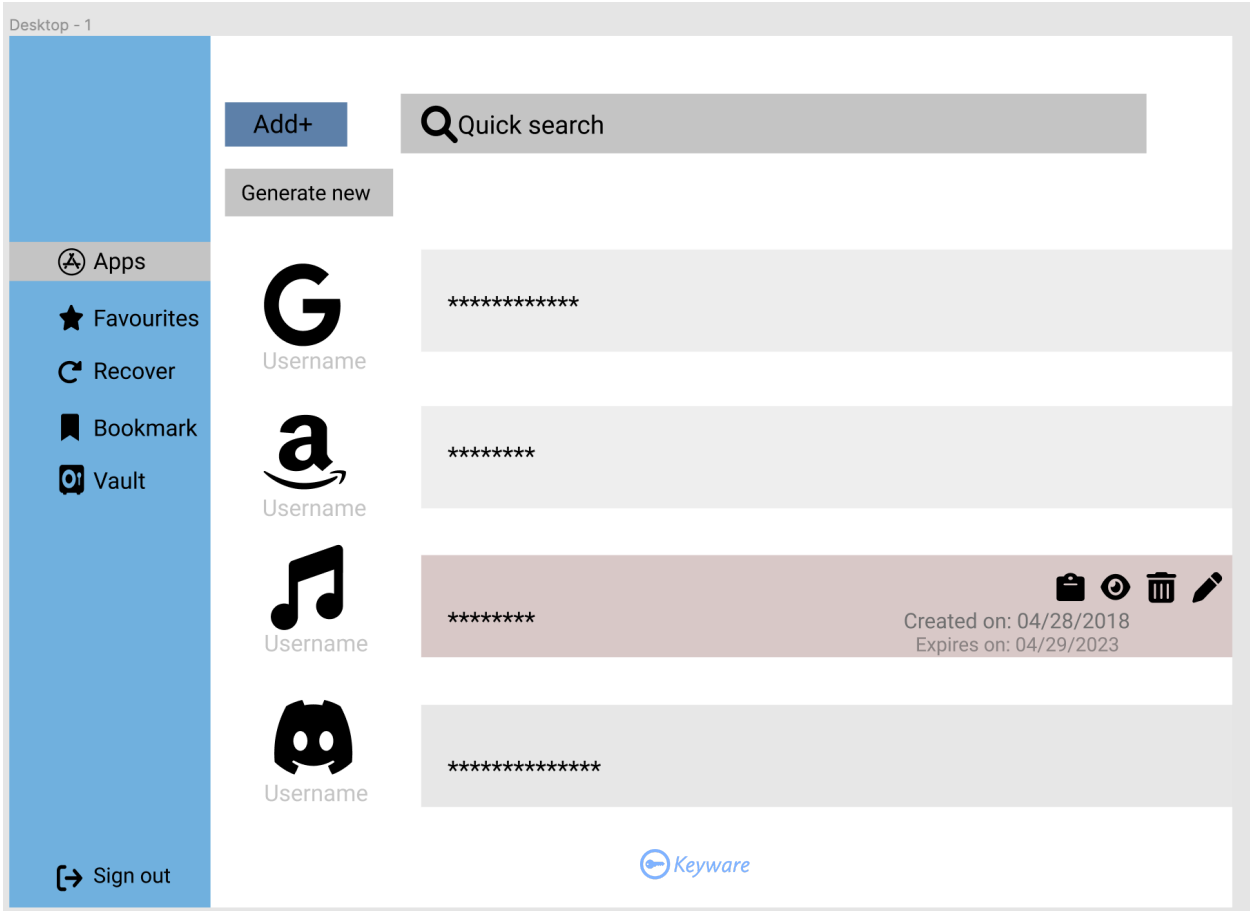
(Security Question Here)

Security Question Answer

Reset Password

[Cancel](#)

d. Main Password Manager



6. Glossary

N/A	N/A
-----	-----