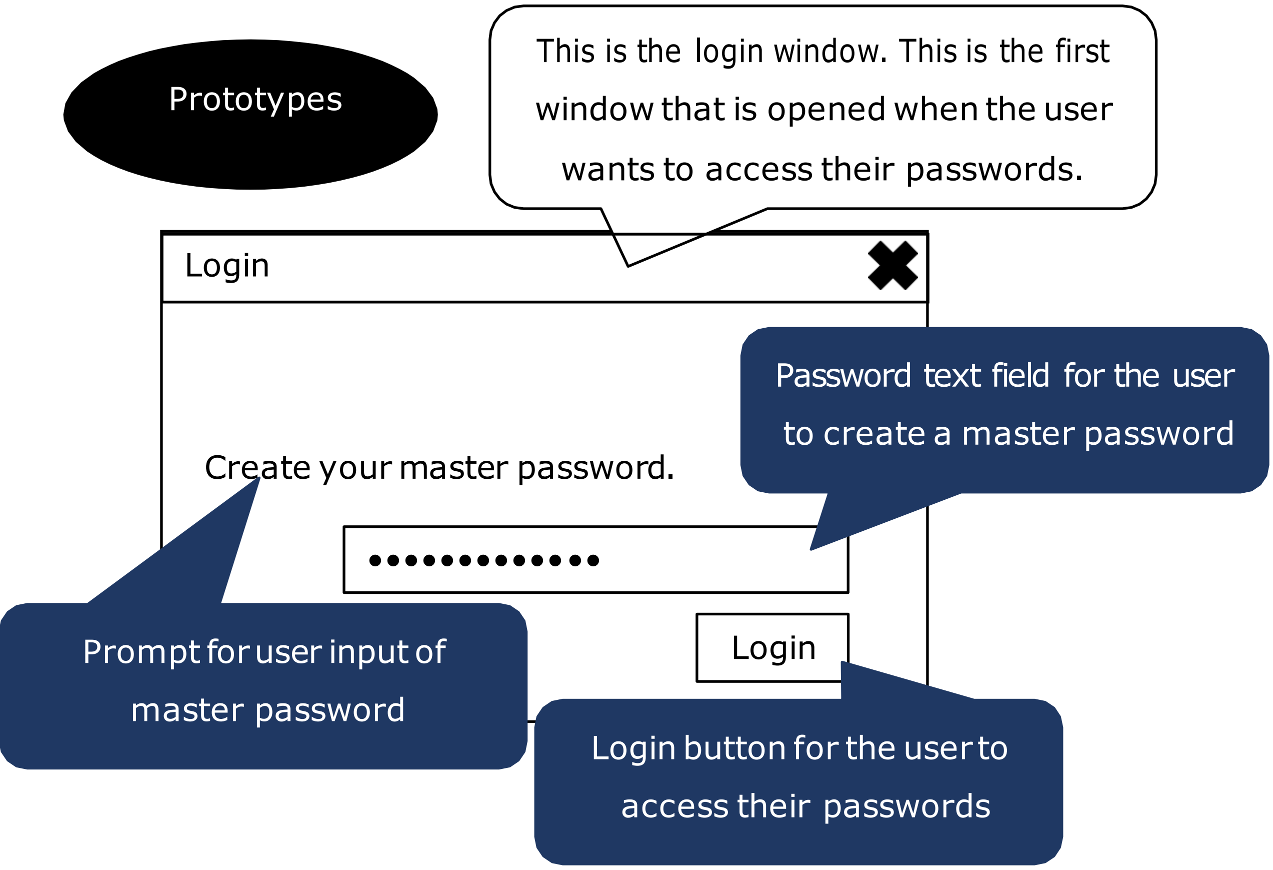
Design

Before beginning to develop the program, I have to employ a clear roadmap and blueprint to make development simple.

Notes:

- Make the application functional and visually appealing simultaneously.
- Comment code to maintain organization.
- Only import packages that are needed.
- Name behaviors and attributes using camel case, and choose names that are useful for recognition of functionality.
- User proper case for naming classes and all caps for constants.
- Use modularity for commonly used code.
- Passwords must be encrypted; the master password should be hashed.

Prototypes

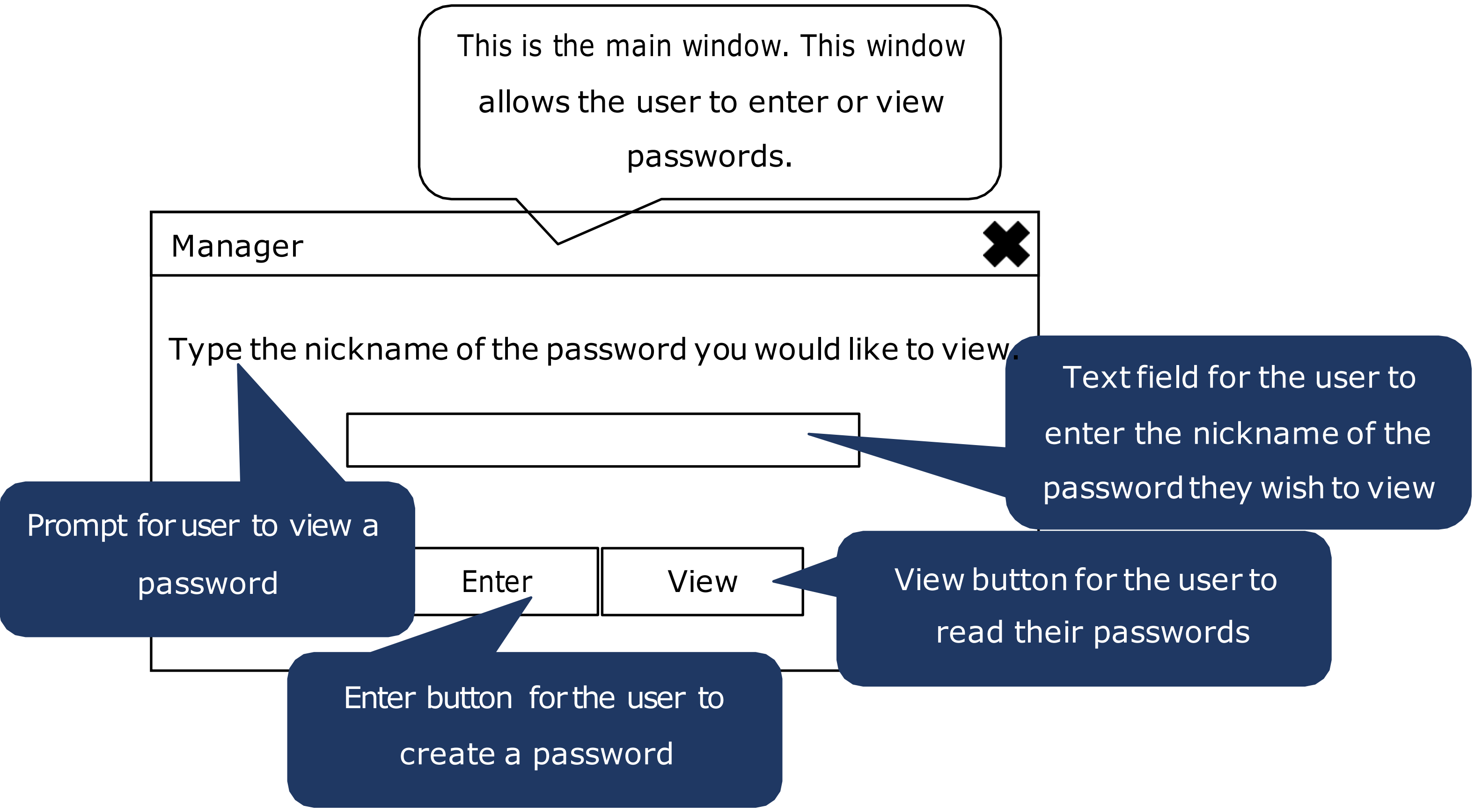


This is the login window. This is the first window that is opened when the user wants to access their passwords.

Password text field for the user to create a master password

Prompt for user input of master password

Login button for the user to access their passwords



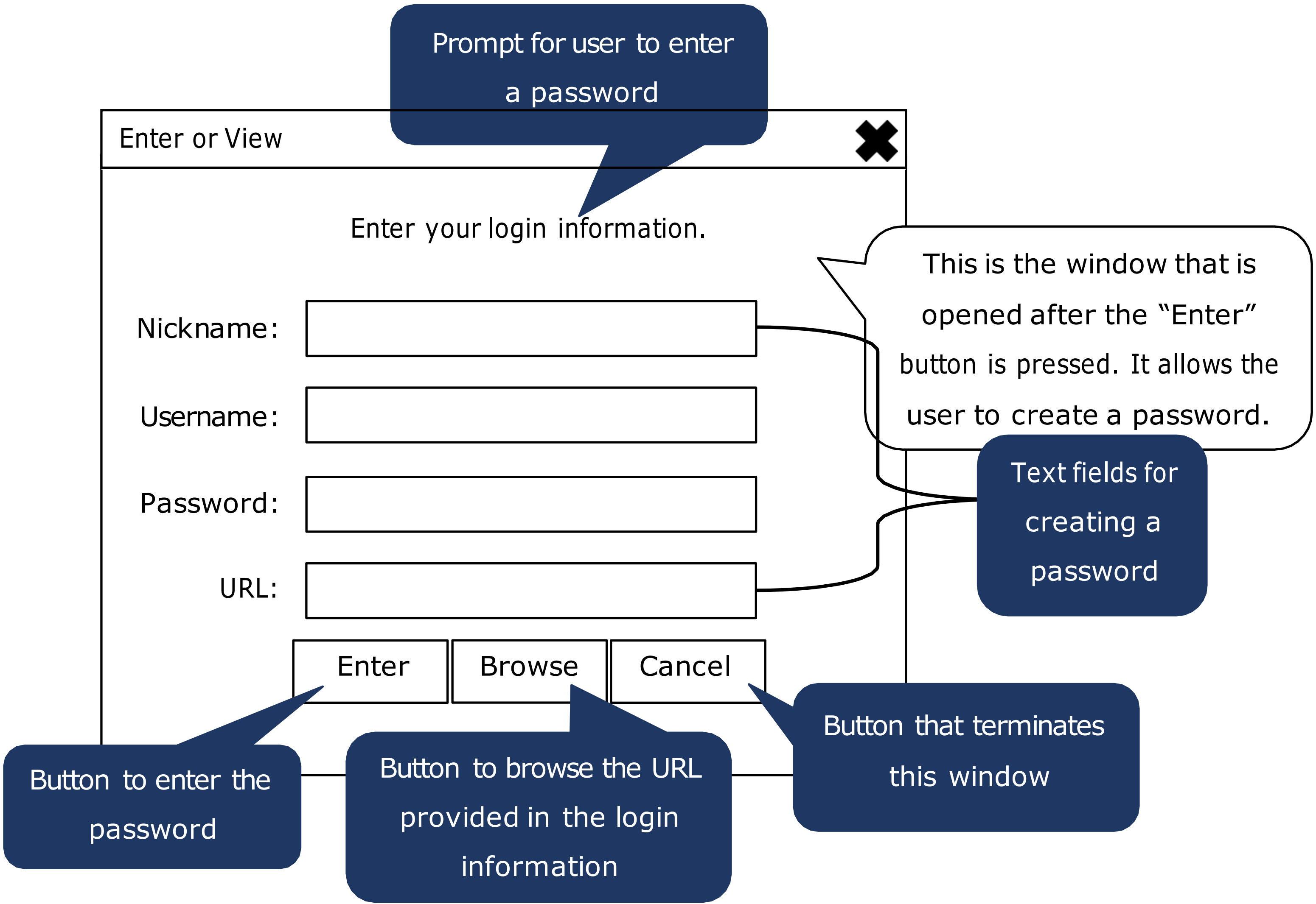
This is the main window. This window allows the user to enter or view passwords.

Text field for the user to enter the nickname of the password they wish to view

Prompt for user to view a password

View button for the user to read their passwords

Enter button for the user to create a password



Enter your login information.

Nickname:

Username:

Password:

URL:

Prompt for user to enter a password

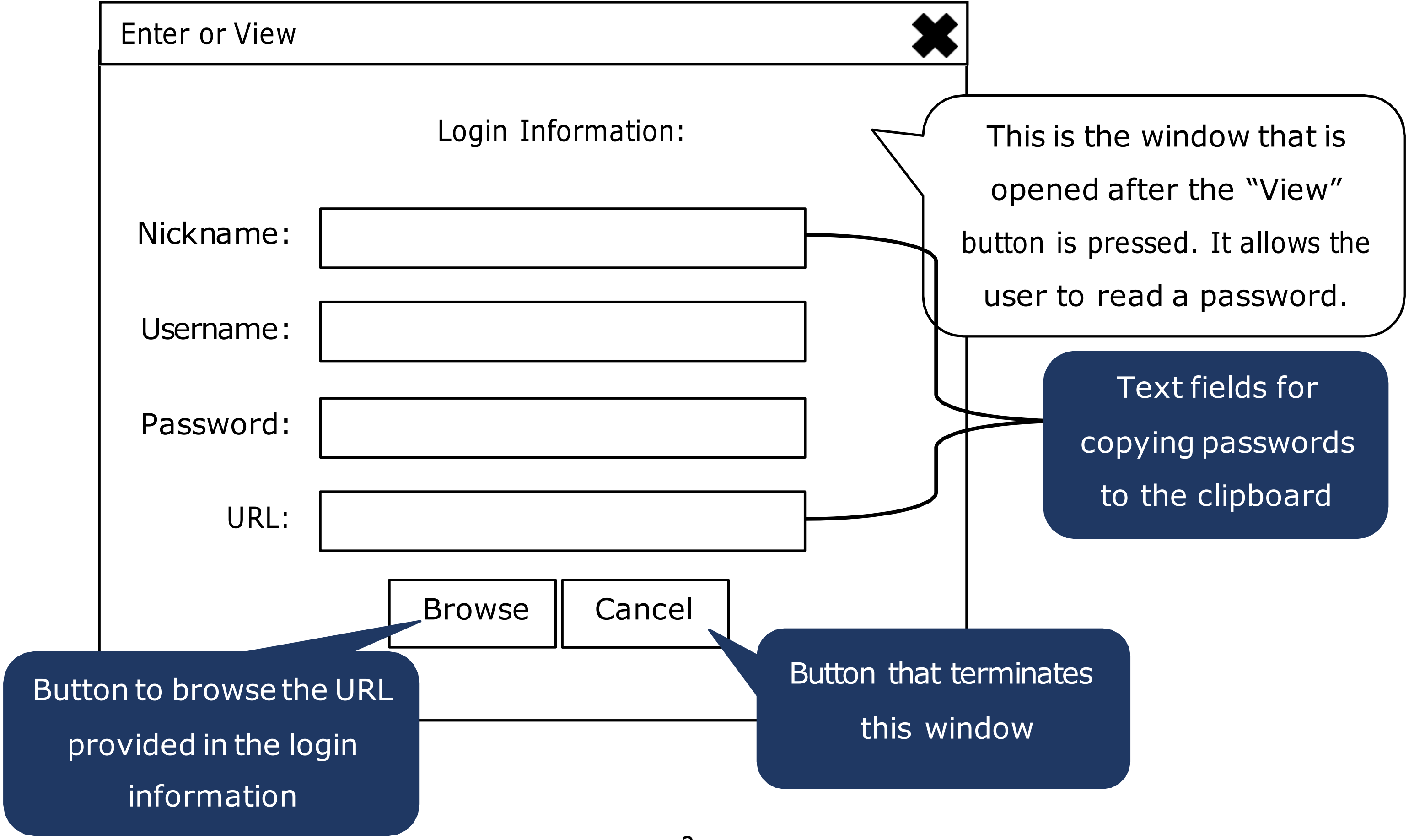
This is the window that is opened after the "Enter" button is pressed. It allows the user to create a password.

Text fields for creating a password

Button to enter the password

Button to browse the URL provided in the login information

Button that terminates this window



Login Information:

Nickname:

Username:

Password:

URL:

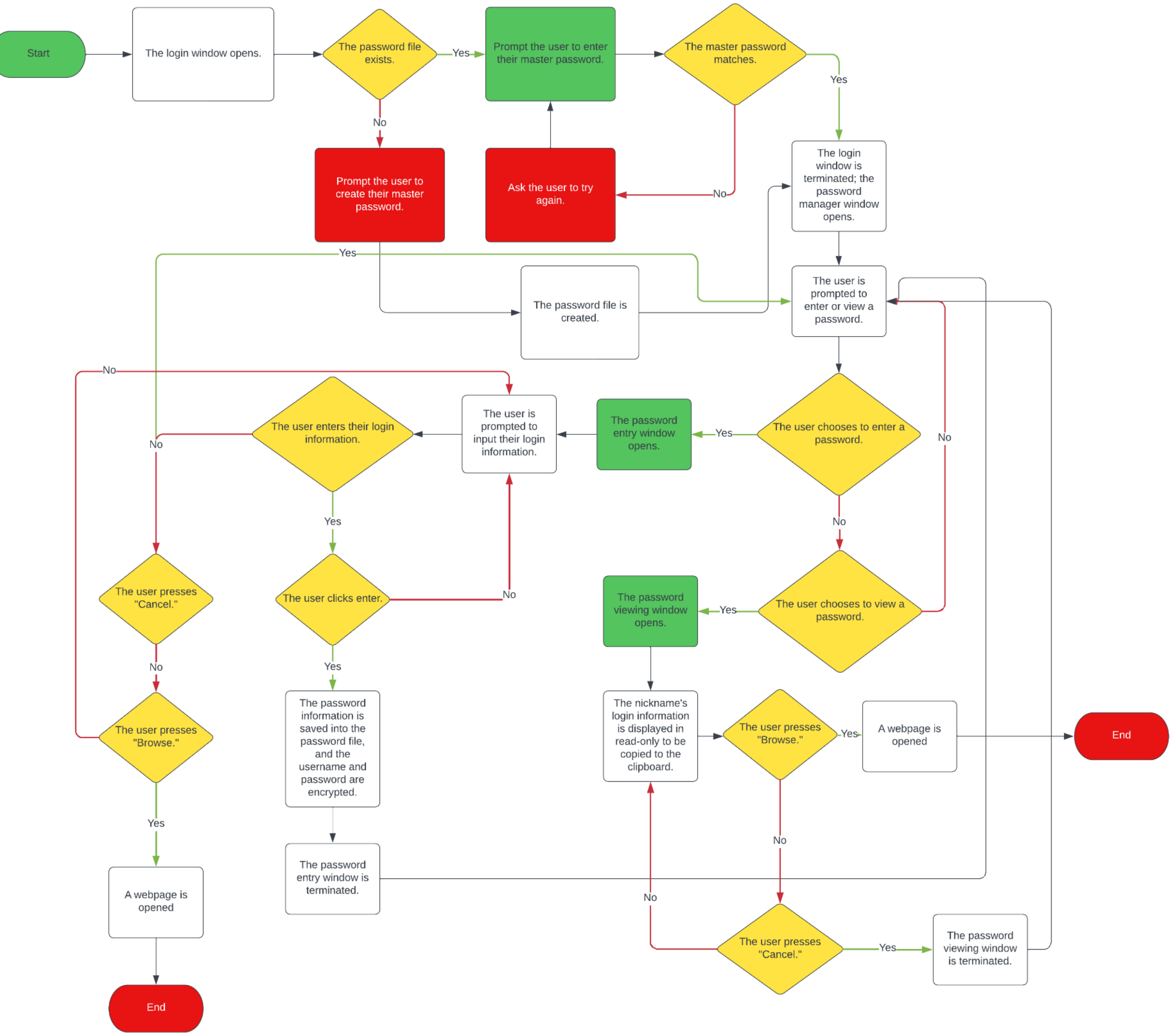
This is the window that is opened after the "View" button is pressed. It allows the user to read a password.

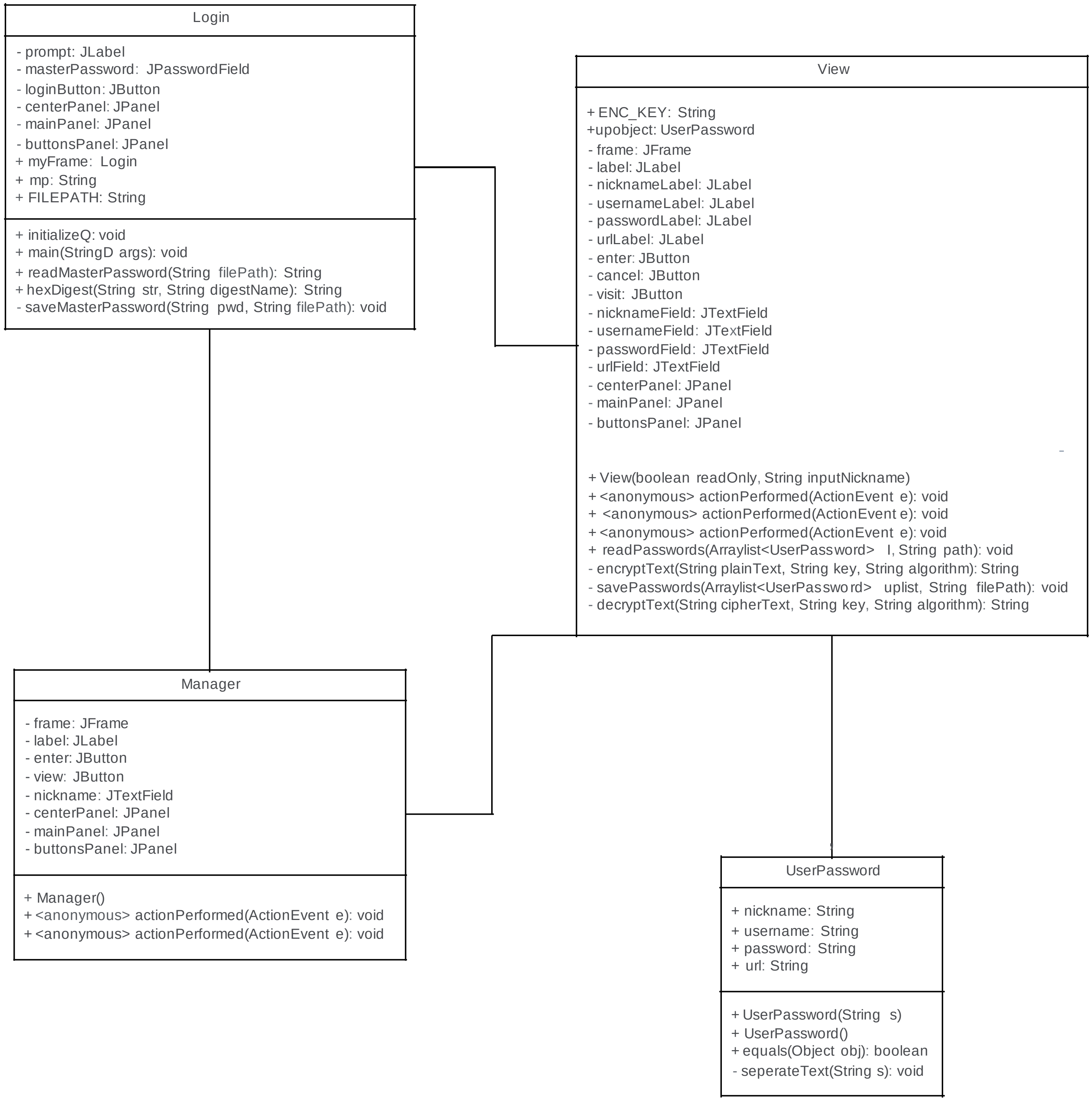
Text fields for copying passwords to the clipboard

Button to browse the URL provided in the login information

Button that terminates this window

System flowchart:


Start
The login window opens.
The password file exists.
Yes
No
Prompt the user to enter their master password.
The master password matches.
Yes
No
Prompt the user to create their master password.
Ask the user to try again.
The login window is terminated; the password manager window opens.
Yes
The password file is created.
The user is prompted to enter or view a password.
No
The user is prompted to input their login information.
The password entry window opens.
Yes
The user chooses to enter a password.
No
The user enters their login information.
No
Yes
No
The user chooses to view a password.
The user presses "Cancel."
The user clicks enter.
No
The password viewing window opens.
Yes
No
Yes
The user presses "Browse."
The password information is saved into the password file, and the username and password are encrypted.
The nickname's login information is displayed in read-only to be copied to the clipboard.
The user presses "Browse."
Yes
A webpage is opened
End
Yes
No
The password entry window is terminated.
A webpage is opened
No
The user presses "Cancel."
Yes
The password viewing window is terminated.
End



Login

- prompt: JLabel
- masterPassword: JPasswordField
- loginButton: JButton
- centerPanel: JPanel
- mainPanel: JPanel
- buttonsPanel: JPanel
+ myFrame: Login
+ mp: String
+ FILEPATH: String

+ initializeQ: void
+ main(StringD args): void
+ readMasterPassword(String filePath): String
+ hexDigest(String str, String digestName): String
- saveMasterPassword(String pwd, String filePath): void

View

+ ENC_KEY: String
+upobject: UserPassword
- frame: JFrame
- label: JLabel
- nicknameLabel: JLabel
- usernameLabel: JLabel
- passwordLabel: JLabel
- urlLabel: JLabel
- enter: JButton
- cancel: JButton
- visit: JButton
- nicknameField: JTextField
- usernameField: JTextField
- passwordField: JTextField
- urlField: JTextField
- centerPanel: JPanel
- mainPanel: JPanel
- buttonsPanel: JPanel

+ View(boolean readOnly, String inputNickname)
+ <anonymous> actionPerformed(ActionEvent e): void
+ <anonymous> actionPerformed(ActionEvent e): void
+ <anonymous> actionPerformed(ActionEvent e): void
+ readPasswords(Arraylist<UserPassword> I, String path): void
- encryptText(String plainText, String key, String algorithm): String
- savePasswords(Arraylist<UserPassword> uplist, String filePath): void
- decryptText(String cipherText, String key, String algorithm): String

Manager

- frame: JFrame
- label: JLabel
- enter: JButton
- view: JButton
- nickname: JTextField
- centerPanel: JPanel
- mainPanel: JPanel
- buttonsPanel: JPanel

+ Manager()
+ <anonymous> actionPerformed(ActionEvent e): void
+ <anonymous> actionPerformed(ActionEvent e): void

UserPassword

+ nickname: String
+ username: String
+ password: String
+ url: String

+ UserPassword(String s)
+ UserPassword()
+ equals(Object obj): boolean
- seperateText(String s): void

Methods in the Login Class:

Name	Access Modifier	Type	Return Type	Parameters	Description
initialize	public	None	void	None	Creates the Login GUI for inputs
main	public	static	void	String[] args	Creates the JFrame for the GUI
readMasterPassword	public	static	String	String filePath	Reads the first line of the file path and returns the value that is there
hexDigest	public	static	String	String str, String digestName	Hashes the given String using the specifies algorithm (SHA-256)
saveMasterPassword	private	None	void	String pwd, String filePath	Saves the master password in the file on the first line

Methods in the Manager Class:

Name	**Access Modifier**	**Type**	**Return Type**	**Parameters**	**Description**
Manager	public	------	None	None	Creates the Password Manager GUI for inputs
actionPerformed	public	None	void	ActionEvent e	Sets up an Event for the "Enter" button
actionPerformed	public	None	void	ActionEvent e	Sets up an Event for the "View" button

Methods in the View Class:

Name	Access Modifier	Type	Return Type	Parameters	Description
View	public	------	None	boolean readOnly, String inputNickname	Creates the Enter or View GUI for inputs
readPasswords	public	static	void	ArrayList<UserPassword> L, String path	Skips the first line of the file and reads and returns the value that is there
encryptText	private	static	String	String plainText, String key, String algorithm	Encrypts the specified String using a key and a specified algorithm (AES)
savePasswords	private	static	Void	ArrayList<UserPassword> UPList, String FilePath	Saves the specified credentials on a single line in a specific file
decryptText	private	static	String	String cipherText, String key, String algorithm	Decrypts the specified ciphered text using a key and a specified algorithm (AES)

Methods in the UserPassword Class:

Name	**Access Modifier**	**Type**	**Return Type**	**Parameter s**	**Description**
UserPassword	public	------	None	String s	Separates the specified String
UserPassword	public	------	None	None	Declares empty "UserPassword" objects to be manipulated
equals	public	------	boolean	Object obj	Returns true or false whether two "UserPassword" objects are equal based on their nickname
separateText	private	None	void	String s	Separates the credentials, which are delimited by the semicolons in the file

Test Plan:

Test #	Test Title	Test Values	Expected Outcome	Criteria Satisfied
1	Program is Run	User runs program.	Login window is displayed.	5 – The user can interact with buttons immediately, and a window is popped up.
2	Master Password Created	User creates a master password.	Password Manager window is displayed.	3 – Master password is included.
3	Password Entry	Enter button is pressed and the user inputs credentials	Encrypted password is saved.	1 – The user can enter passwords.
5	Password Viewing	View button is pressed after nickname is inputted.	Credentials of the specified nickname is displayed.	1 – The user can search for passwords.
6	Password Copy and Paste	Highlight the password and copy and paste	The password will be saved to your clipboard.	7 – The user can copy and paste information
7	Web Browsing	Browse button is pressed.	The website that the user entered is displayed.	6 – The user can visit a website through the application.
8	Password Remembrance	Re-Open the application, login, click view, and input a nickname.	The specified nickname credentials are displayed.	4 – The passwords are viewable when the application is closed.

Test Plan (con't):

Test #	Test Title	Test Values	Expected Outcome	Criteria Satisfied
9	Password Encryption and Hashing	Open the "Passwords.txt" file.	The first line is hashed and the lines following are encrypted.	3 – The master password is hashed. 2 – The passwords are encrypted when stored.