Website Manager

Overall Goal:

To create a program (GUI) where you can store usernames and passwords, and then with one click, be able to automatically open a variety of programs for some activity (i.e., grading window from skyward pops up without having to navigate there).

Use Cases:

Use case	Accessing Programs
Actor	User - Teachers
Flow	When opening the program, a tab will appear including options to access Skyward gradebook, skyward attendance, create new assignments on teams, email/outlook, frontline absence management, frontline – evaluation.
Alternating flow 0	If the teacher clicks on Skyward gradebook, the program opens chrome and automatically logs into Skyward. Without user input it will navigate to Skyward gradebook.
Alternating flow 1	If the teacher clicks on Skyward attendance, the program opens chrome and automatically logs into Skyward. Without user input it will navigate to Skyward attendance.
Alternating flow 2	If the teacher clicks on creating new assignments on teams, the program opens teams and logs in if necessary. Without user input it will navigate to assignments on teams.
Alternating flow 3	If the teacher clicks on email/outlook, the program opens outlook in chrome and logs in if necessary.
Alternating flow 4	If the teacher clicks on frontline absence management, the program opens frontline and logs in if necessary. Without user input it will navigate to absence management on frontline.
Alternating flow 5	If the teacher clicks on frontline evaluations, the program opens frontline and logs in if necessary. Without user input it will navigate to evaluations on frontline.

Use case	Creating new links
Actor	User - Teachers
Flow	When opening the program, a tab will appear which will include the option to create a new link. The user clicks on that option, and it brings up a new tab in which the user can input an app or website they want access to along with username or password. After the user inputs that information it returns to the main program tab in which the new quick link is added.

Project Restrictions:

Take up as little space as possible, doesn't take up all space on the hard drive

Computer stays usable before and after the main part of the app runs (opening all 3 windows)

Must be more efficient/take less time than manually opening programs and signing in to them

External Resources:

Websites that Mr. Danaee needs to sign into

JFrame to make a GUI

Google chrome to open any websites

Robot class and/or Selenium to automate moving through webpages

Passwords stored in a local folder (possibly encrypted), so the passwords are not shared on any cloud service

Teams API that allows us to navigate through the application

Stored as addresses in an external text document

Requirements

Must have:

- The ability to store usernames and passwords.
- Ability to access and open Google Chrome
- Ability to automatically log into specified websites
- Ability to automatically navigate through webpages

Nice to have:

- Ability for the user to add new quick links to the program

Time permitting

- Ability to open Multiple bowsers from one quick link
- Keeping everything to one window when applicable
- Encryption on local folder storing usernames and passwords
- Visual security precautions when inputting passwords (changing password characters to dots)

Clarifications from Mr. Danaee:

Quick links to the following would be required:
Skyward - gradebook
Skyward - attendance

Frontline - evaluations
Pearson's Mastering Biology – login
AP Classroom – login

Wishlist: The ability to add new quick links myself as needed. (i.e. AP Classroom, Mastering Biology, ExploreLearing, etc)
Use Google Chrome

Changes Due to Complications

We had to change to using the student view of skyward, as none of us had access to the teacher's view in skyward.

Accessing Outlook and Teams was cut because it wasn't feasible due to security

Accessing Frontline was cut because it wasn't feasible in the timeframe we had to work on the project

Original Timeline:

First draft of spec by 10/22/21 - meet in final project

Skeleton APIs by 10/25/21 - took longer to put together than expected

Sorted in order of hierarchy:

1. Quick Links (Atharv) 10/27/21

2. Text File Manager (Thys Vanderschoot) 10/29/21

3. Create Links (Samuel Ferreira) 10/2/21 & Desktop Manager (Marko) 10/2/21 - newQuickLinks took longer to code than expected, Desktop manager ran into complications with selenium

4. GUI (Leane) 11/3/21

5. Runner (Atharv) 11/3/21

Completed Project by 11/05/21 - the entire project was pushed back a week by Mrs. Kankleborg

Timeline:

First Draft of Product Specification completed by October 22, 2021

Marko worked on downloading selenium October 26 – 27

Atharv worked on the QuickLinks constructor and accessor methods October 26-27

Leane worked on GUI visuals October 27 – 28

Atharv worked on the QuickLinks mutator methods October 27-28

Marko configured eclipse project to maven project October 28 – 29 plans had to change plans because code couldn't compile on school computers

Thys divided FileManager into two classes, created new skeleton API for both

Skeleton APIs completed by October 29

Marko finding alternative to Maven in Selenium November 1 - 8

Leane worked on GUI newButton November 1

Thys worked on default file creation November 1-2

Leane worked on GUI createNewLink November 2 - 5

Thys worked on reading file with getNextSet November 3 - 4

Samuel worked on NewQuickLinks from November 1 – 4

Atharv worked on addLink in LinkManager on November 4

Thys worked on editing file through writeToFile November 5 - 8

Atharv worked on getLink in LinkManager on November 5

Leane worked on GUI createNewLink November 7 - 9

Samuel worked on LinkManager addCredentials from November 5 - 12

Marko worked on DesktopManager November 9 - 13

Thys finished constructor for FileManager November 9

Samuel completed and merged NewQuickLinks November 11

Thys started and submitted methods in FileManager and LinkManager November 12

Samuel completed and merged LinkManager addCredentials November 12

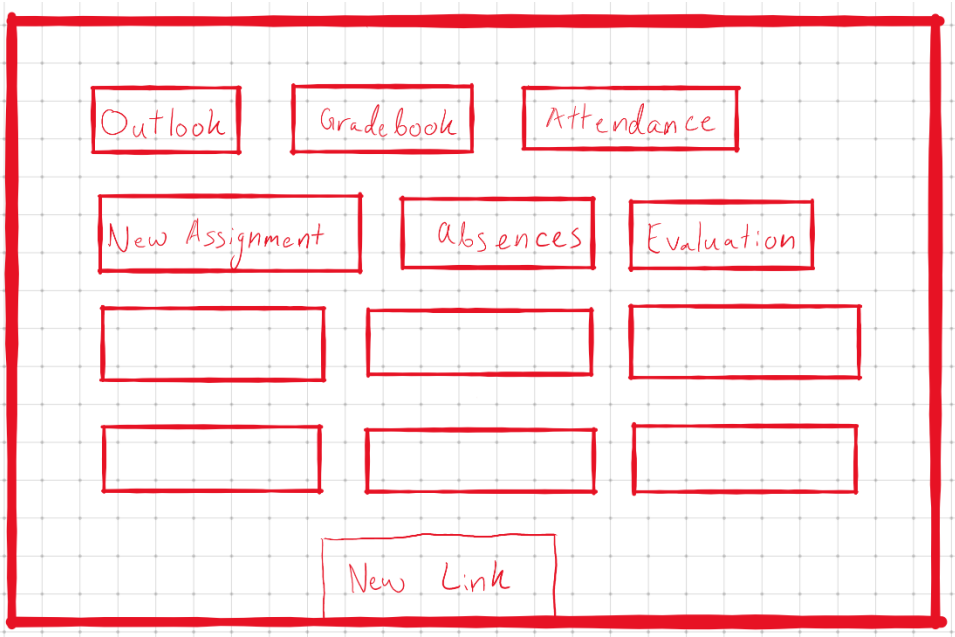
Leane worked on adapting GUI to work dependently November 12 – 15

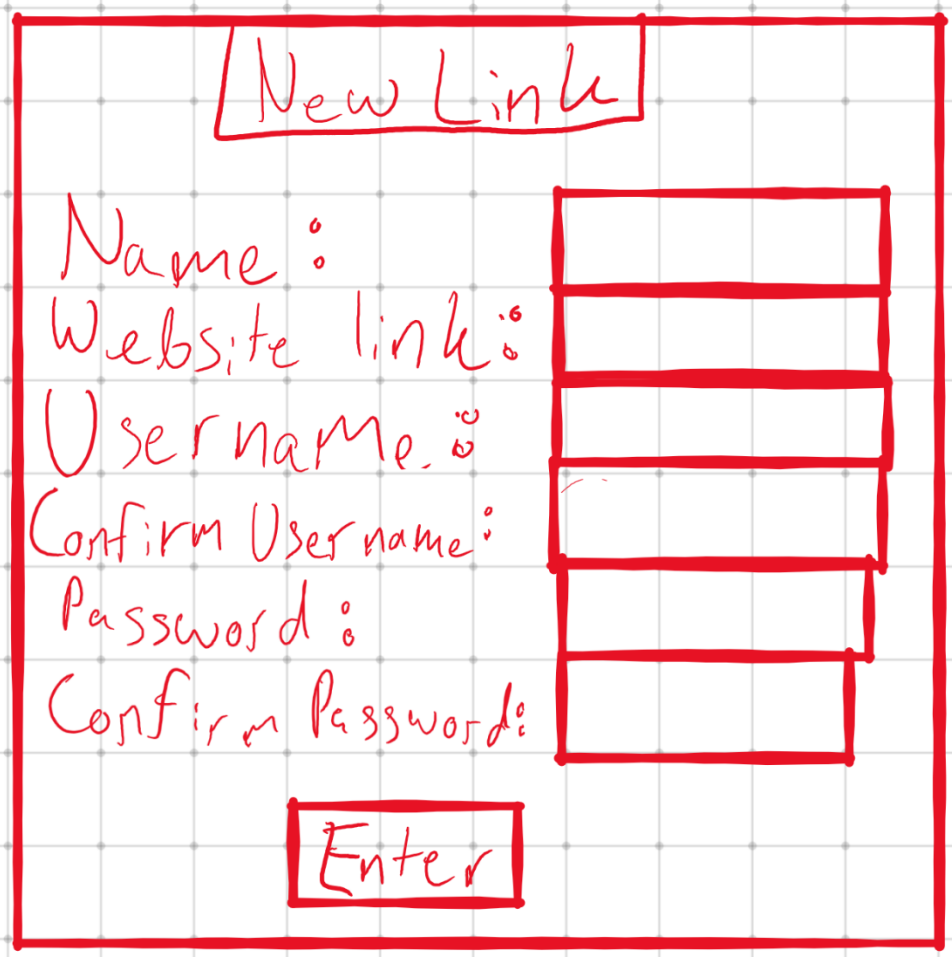
Atharv worked on making GUI work with LinkManager November 15

Thys and Marko worked on GUI so that it interacts with DesktopManager, NewQuickLink, and FileManager November 15th

Project completed November 15th

Initial Layouts:


Outlook
Gradebook
Attendance
New Assignment
Absences
Evaluation
New Link


New Link
Name:
Website link:
Username:
Confirm Username:
Password:
Confirm Password:
Enter