

Design Studio 3

WayBackWhen



Subject	Design Studio 3							
Class	Informatics 121: Software Design 1							
Professor	Alberto Krone-Martins							
Date	December 3, 2021							
Team	Group 3							
Members	<table border="1"><tr><td>Claire Wodzien</td><td>Htoo Zay</td></tr><tr><td>Mairie Espiritu</td><td>Ian Maynard</td></tr><tr><td>Joshua Hsin</td><td></td></tr></table>		Claire Wodzien	Htoo Zay	Mairie Espiritu	Ian Maynard	Joshua Hsin	
Claire Wodzien	Htoo Zay							
Mairie Espiritu	Ian Maynard							
Joshua Hsin								

Design Studio 3 WayBackWhen	1
Overview	3
Application Design	3
Feature Comparison:	3
Audience	4
Stakeholders	5
Goals	5
Constraints	6
Assumptions	7
Decisions	7
Ideas	7
Interaction Design	8
Goals	8
Constraints	10
Assumptions	11
Decisions	11
Ideas	11
The Average User and Their Personas	12
Storyboarding the process and reasons	13
UI Mockup and Prototype	15
Architecture Design	18
Pseudocode - website	18
Pseudocode - browser extension	21
UML	24
Appendix	25

Overview

WayBackWhen is a digital archive website with a browser extension that allows users to have a personal collection of website archives that they want to have saved. WayBackWhen is made to satisfy the needs of different audiences: companies, universities, law firms, or your everyday person. WayBackWhen saves the interactive snapshots (with navigation and links between the pages fully functional similar to that of the original) of the desired website with the goal of allowing companies to keep track of changes in their websites, saving personal pages for the future, or to bring up comparisons of the same website during two different times. Additional features like saving, favoriting, and downloading snapshots, and the ability to separate saved snapshots into different collections will also give users a more personalized experience compared to our competitors. WayBackWhen is made to be intuitive with the browser extension allowing users to easily access most of the features through simple clicks without needing to leave their website of choice.

Application Design

Feature Comparison:

	archive.today	Wayback Machine	stillio
Timeline Format	Gallery Timeline	Calendar Timeline	Gallery Timeline, List View, Domain View, and tags
Viewing Technique	Interactive Snapshot with redirection to url when user clicks on a link	Webpage Simulation	Non-Interactive Snapshot
Snapshot technique	User created snapshots, viewable to all users - public snapshots	Work with libraries and other organizations through Archive it program to identify important websites and captures upon changes User can also upload a capture of a current webpage	User controls time intervals of webpage capture from time of webpage saved into the future, user can only see own snapshots and not those of others - private snapshots

Saving technique	Bookmark on your computer	Bookmark on your computer	Save urls and notify user by email on new Snapshots, create tags for user to group and find saved urls more easily
Price	Free	Free	Different price points of subscriptions with different available features



Audience

- Universities
 - Can understand how their catalogue has changed
- Law Firms
 - Can check history of laws that were changed through the time

- Businesses
 - Allows the business to view the history of their external image they portrayed through the website.
- Individuals
 - Can save and view family members online histories, blogs
 - Keep track of and view histories of different web pages

Stakeholders

- Investors
- Website owners
- Data Storage company
- Web crawling software maker
- Government
 - Grants

Goals

- Program is a website that works with a browser extension
- **Website**
 - User can create a new account
 - User can link their account to their Facebook or Google accounts
 - User has an account which holds saved snapshots of websites (fully interactable website)
 - User can choose to remain logged in
 - User can save snapshots
 - User can create and view different collections of snapshots
 - When saving snapshots, user will be asked which folder they want the snapshot to be saved
 - User can delete collections of snapshots
 - User can delete saved snapshots from collections
 - User can view saved snapshots
 - User can save and view web urls (same place as saved snapshot collections)
 - Saved to default collection “favorite urls”
 - User will be notified by email when new snapshots are made of web url
 - User cannot make new collection for web urls
 - User can remove from favorite urls
 - User will no longer receive email notifications
 - Snapshots will be viewable in a Calendar format
 - Days on the Calendar with snapshots will be green, other days will be white
 - Search bar on website that can be used to look up web urls
 - User can view past snapshots as a interactive webpage simulation
 - User can navigate snapshot as normal webpage
 - In simulation, user can navigate forward and backwards to snapshots of different times

- User has option to exit from webpage simulation at any point
- User can download a snapshot as a JPEG, PNG, PDF, or HTML/CSS file
- Top 2000 web urls automatically have one snapshot taken a day
 - Users can create more snapshots
- User can create and save a current snapshot of a web url
- **Browser extension**
 - User can create a new account
 - User can link their account to their Facebook or Google accounts
 - User logs in once on browser extension and remains logged in forever
 - User must be logged in to use browser extension
 - User can save snapshots of web urls that have snapshots
 - Browser extension can be turned on and off
 - User can see all past snapshots of websites in a gallery format or list view
 - User can view past snapshots as a interactive webpage simulation
 - User can navigate snapshot as normal webpage
 - User can create split screen and view two code snapshots of a website at once
 - On each simulation tab, user can navigate forward and backwards to snapshots of different times
 - User has option to exit from webpage simulation tabs at any point
 - User can save web urls to default collection “favorite urls”
 - Saved urls will notify user by email when new snapshots have been made
 - Users can unsave web urls
 - User can download a snapshot as a JPEG, PNG, PDF, or HTML/CSS file
 - User can create a code snapshot of a webpage
 - Uploaded to database and to default personal folder
 - All users can create snapshot of a webpage every three hours
 - User can create code snapshots of web pages not in database
 - User can see gallery view or list view of snapshots of a web url assuming there are snapshots of it
 - Contains a short tutorial upon first use
 - Details different modes of the browser extension
 - How to save and unsave web urls, switch between gallery and list view, create snapshots, save snapshots to collections, and open multiple snapshot simulations
 - Explains three hour rule among all users for snapshots of a website
 - User can download a snapshot as a JPEG, PNG, PDF, or HTML/CSS file

Constraints

- The user shall not be able to select dates that do not exist for the website in question
- The date must be past or current, must not be a future date
- The user must know the URL for the website they are searching for
- The user must have an efficient enough internet connection to not create a timeout error
- There must be efficient storage on the server/ cloud storage to save the websites
- Any private and/or personal data must be excluded and/or omitted from the saved file

- User submissions can only occur every three hours, while the automated saves happen once a day
 - The automated saves will only be ones that a user has previously has saved

Assumptions

- We assume that the user will know how to use a search feature to find the website they are looking for
- We assume that the user will have access to the internet
- We assume that the user will be able to find the website they are looking for
- We assume we have access to the website we are saving into our time capsule
- We assume we are following the most current laws and regulations in regards to the internet and privacy.
- We assume that we are able to grab the HTML and any source code associated
- We assume the user can add plugins on their web browser

Decisions

- Originally decided on only browser extension for easy user access at any point
 - However, decided to also have website so user could view saved webpage and saved snapshots collections
- Website will use Calendar view for snapshots and browser extension will use gallery and list view
- Limit splitscreen simulation tabs to two
- Allow the user to navigate snapshots as if website exists
- User can navigate backward and forward through time to different snapshots on a snapshot simulation
- Saved url default collection will double as email notification list

Ideas

- Browser extension that pops up (similar to honey) when the user is on a website that allows them to open up snapshots of the previous versions
- Allows users to save up their own snapshots
- Side by side comparison where users will be able to put two different versions of the webpage
- Back and forward button to go through the time
- Let the user download the snapshot in different formats
- Every time a website snapshot is updated user have an option to get an email
- Calendar view vs gallery and list view
- Email notification list

Interaction Design

Goals

- **Website**
 - User can make a new account or log in
 - On login on to website, user has the option to remain logged in
 - User can log out at any time by right clicking on account button on any non account page, then clicking log out
 - Or clicking setting, then log out on the account page
 - User can view collections by clicking account button to navigate to the account page showing “favorite urls” collection, “favorite snapshots” collection, and other user created snapshot collections
 - User can click on “favorite urls” to view all saved urls
 - User can remove urls from this collection by right clicking on url, then clicking delete
 - User will then no longer receive email notifications when new snapshots are made of web url
 - User can delete collections by right clicking on collections (except for default “favorite url” and “favorite snapshots” collections), then clicking delete
 - User can view snapshots in collections by clicking on appropriate collection
 - snapshots in collections can be deleted the same way as collections
 - User can create new collections by clicking create new collection, giving the collection a name, and clicking create collection
 - On user search, the associated Calendar view will show up if the url is in the software’s database
 - Otherwise, user will be notified that the search has failed and the url has no snapshots that have been created before
 - After searching for a web url, a Calendar showing days with snapshots will be displayed (days that are green)
 - After hovering over a day with snapshots, hyperlinks with snapshots times will show up
 - After clicking on link, snapshot simulation will appear
 - Snapshot simulation
 - User can browse like normal website
 - User can click forward and back buttons to go to the next most recent and next least recent snapshots in time, relative to the current one in the simulation
 - User can delete snapshots by clicking on x button in top left corner of each tab
 - User can save urls and snapshots of webpages when logged in

- User can save urls by clicking save website button that will show up at the top of a page after successful search - saved to collection “favorite urls”
- User can save snapshots by clicking next to snapshot link showing the time of snapshot after hovering over a day on the Calendar with snapshots
- User can download snapshots by right clicking on url link after hovering over day on Calendar
 - Download button will appear
 - After clicking on download, user will be asked which format to save in
 - After confirming, web snapshot will be downloaded to user system in selected format
- Top 2000 web urls automatically have one snapshot taken a day
 - Users can create more snapshots
- User can create snapshots of a current web page by navigating to the create snapshot section at the bottom of any page of the website
 - User enters exact name of web url in text field
 - User clicks create
 - If web url is a valid url
 - If website has been saved in last three hours by a user
 - Error: Inform user that snapshot of that web url has already been created within the last three hours
 - Else
 - Create new snapshot of web url at current time, upload to database, then prompt user which collection they would like to save to
 - Save to collection user selects
 - If web url is not valid
 - Inform user that web url does not exist
- **Browser Extension**
 - User can make a new account or log in
 - After first login on browser extension, user remains logged in indefinitely
 - Browser extension shows up as a button in the bottom right corner of a page when browser extension is being used
 - Actions
 - (if there are no snapshots of the url) User can create a code snapshot of the webpage at the moment
 - Button will appear as a plus
 - Snapshot will first be uploaded to database, then be saved to snapshot folder of users choice
 - Extension will then transition to gallery view
 - (if there are snapshots of the url) User can reveal gallery view of snapshots of web page
 - Button will appear as standard button

- User can switch between a gallery view and list view of snapshots when viewing snapshots of a web url by clicking options near the top
- User can also create snapshots on web url with existing snapshots by clicking create new snapshot button on browser extension
 - Snapshot will first be uploaded to database, then be saved to snapshot folder of users choice
- User can save urls and snapshots of webpages when logged in
 - User can save urls by clicking the save website button above the gallery or list view of snapshots
 - User can unsave url if url is already saved
 - Save website button will be an unsave website button, and web url will be removed from default collection “favorite urls”
 - User will no longer be notified by email about new snapshots of this url
 - User can save snapshots by clicking the pin symbol
 - In the top left corner of snapshot preview image if in gallery view
 - Next to the hyperlinks of snapshots showing snapshot time if in list view
- User can view snapshots by clicking on snapshot images or links - snapshot simulation
 - User can browse like normal website
 - User can click forward and back buttons to go to the next most recent and next least recent snapshots in time, relative to the current one in the simulation
 - User can create up to two snapshot simulation by right clicking on a snapshot in the extension and clicking create new tab, which will split the screen in half
 - User can delete snapshots by clicking on x button in top left corner of each tab
- User can download snapshots by right clicking on snapshot image in gallery view or right clicking on url link in list view
 - Download button will appear
 - After clicking on download, user will be asked which format to save in
 - After confirming, web snapshot will be downloaded to user system in selected format

Constraints

- User must meet minimum upload speed to create and upload snapshots to database
- Any web url must only be allowed one snapshot every three hours among all users
- There must be efficient storage on the server/ cloud storage to save the websites
- Any private and/or personal data must be excluded and/or omitted from the saved file

- User must be logged in to use the software for both website and browser extension
- Software will follow general design principles and conventions - for buttons (save, create), calendar, search, etc

Assumptions

- User will have the browser extension downloaded to receive the whole capabilities of the software
- User is an average internet user that understands how to browse websites, use simple extensions, and can understand how to use website through exploration
 - User recognizes general software design conventions
 - General design principles and conventions will help to simplify this process
- User will have appropriate upload and download speeds to be able to create snapshots and navigate through snapshots in an appropriate amount of time
- User will have browser extension turned on

Decisions

- User can right click, then click delete, to delete saved items on account page - snapshots, collections, urls
 - Default collections cannot be deleted
- X button to escape snapshot simulation tabs immediately
- Save button to save web url on both website
- For browser extension
 - For web urls with existing snapshots
 - Button will be a plus
 - For web urls without existing snapshots
 - Button will be a standard button
- Pin symbol representing save

Ideas

- User can right click on saved snapshots, collections, urls, then click delete, to delete corresponding item on website
- X button to escape snapshot simulation tabs immediately
- Save button to save web url on both website and browser extension
- Visual differentiation between actions for web urls with existing snapshots and web urls without snapshots (for browser extension)
- Pin symbol representing save

The Average User and Their Personas

For the personas, we decided to go with a working class person and someone who is a bit younger and inexperienced in the work field. Jane is a professor and focuses a lot on her kids

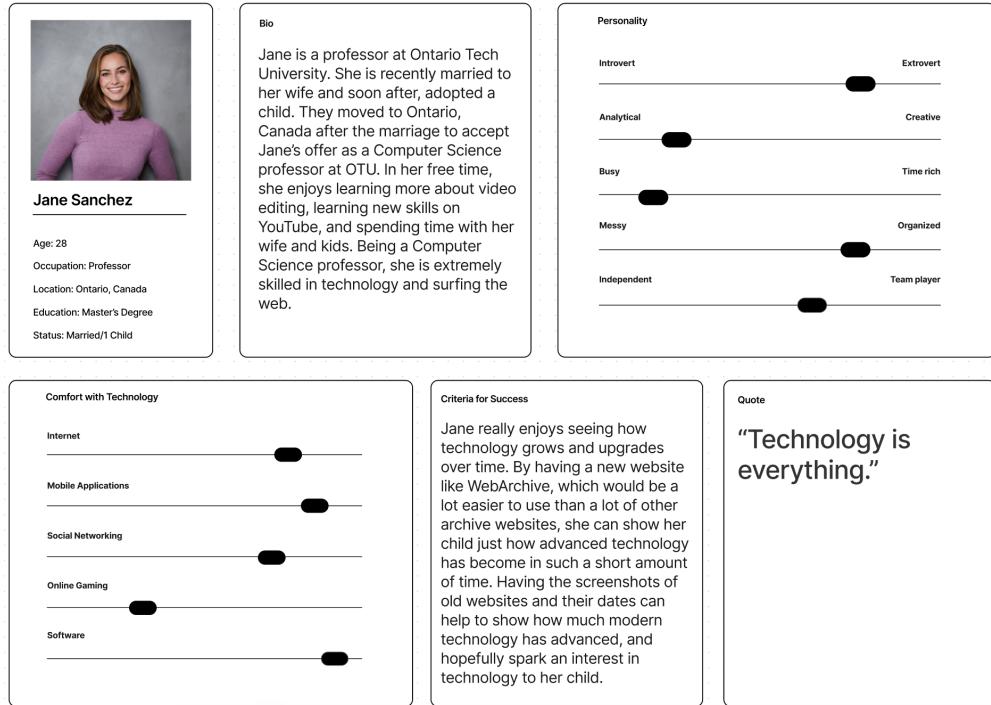


Figure 1: Jane Sanchez - Persona

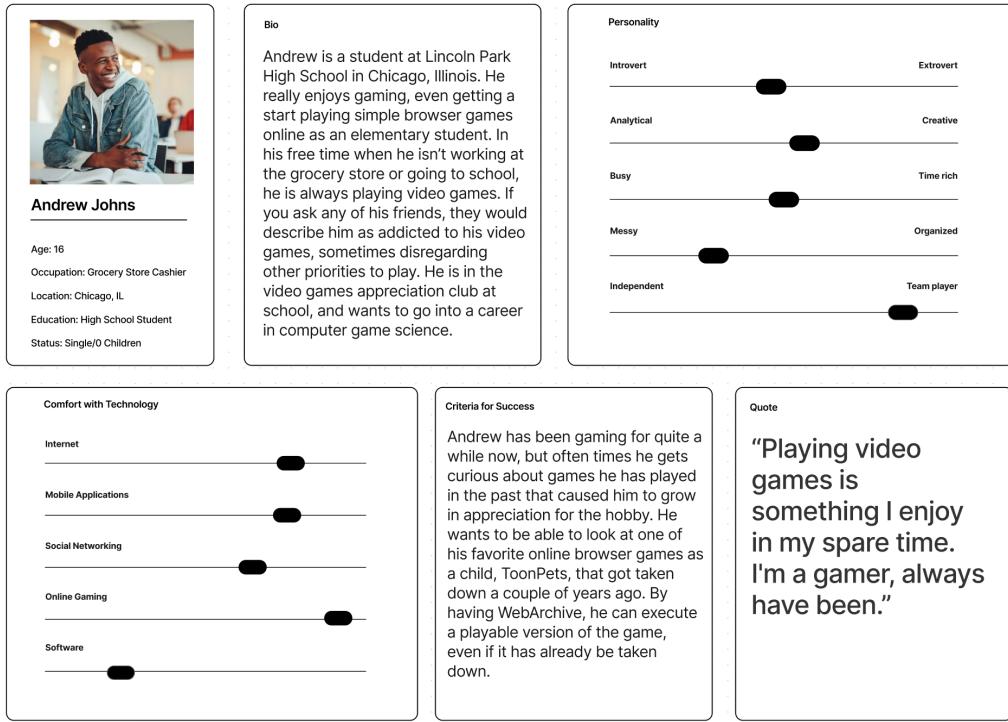


Figure 2: Andrew Johns - Persona

Storyboarding the process and reasons

Below, we show the two different types of users using WayBackWhen through different means (website and browser extension).

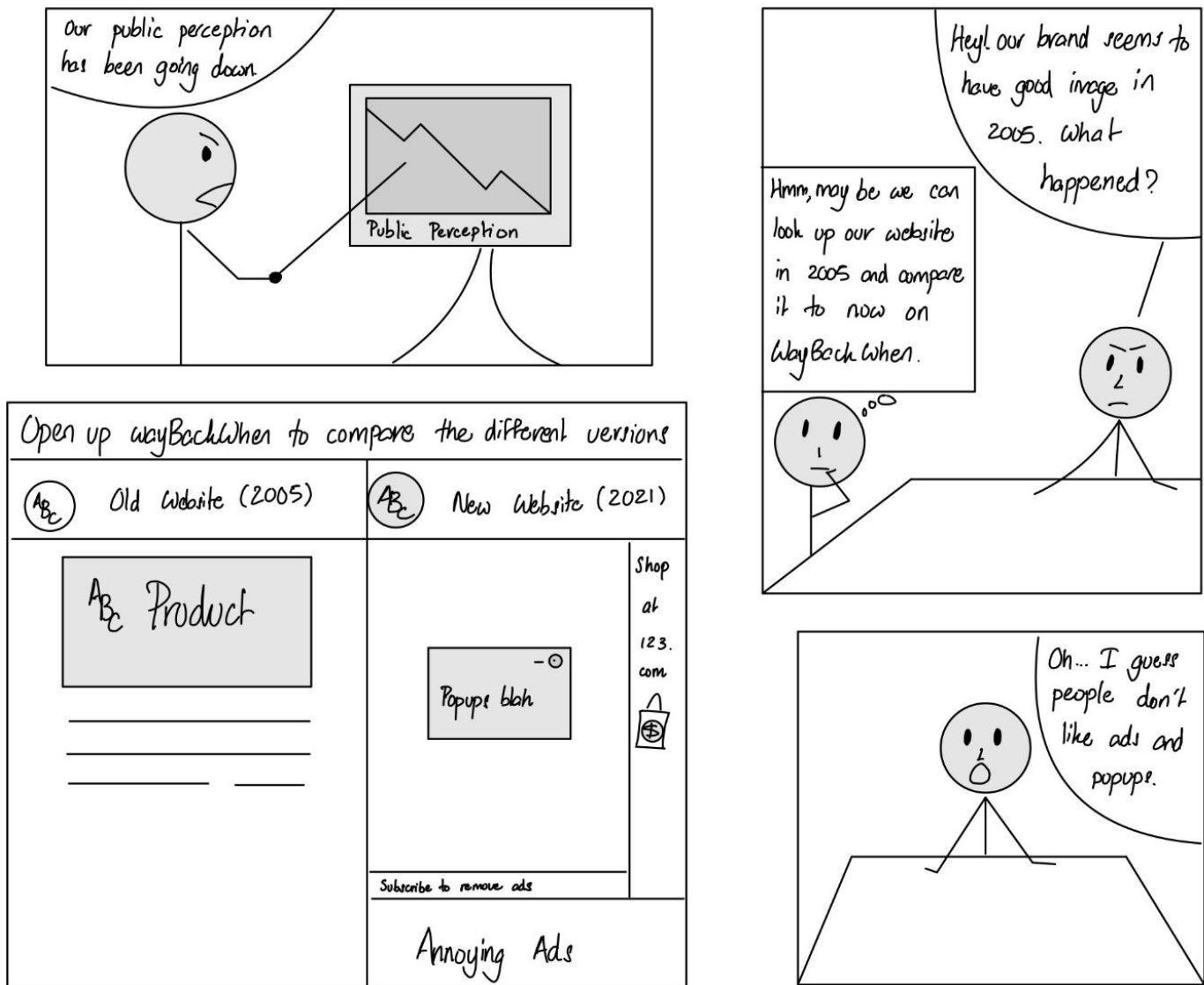


Figure 1: A business comparing through website

The storyboard above shows the reason why a business might be interested in using WayBackWhen and the benefits it could bring. It also shows how they can compare an old version of their website to the new one side by side through our website.

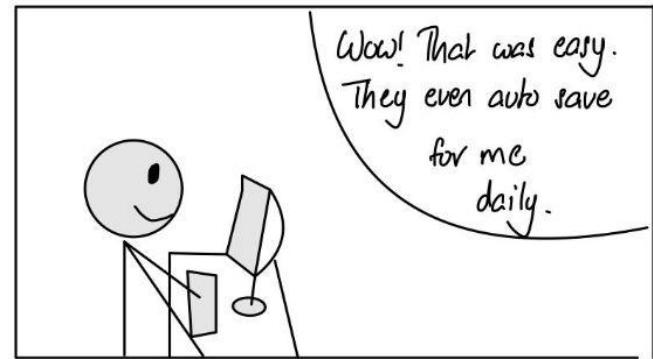
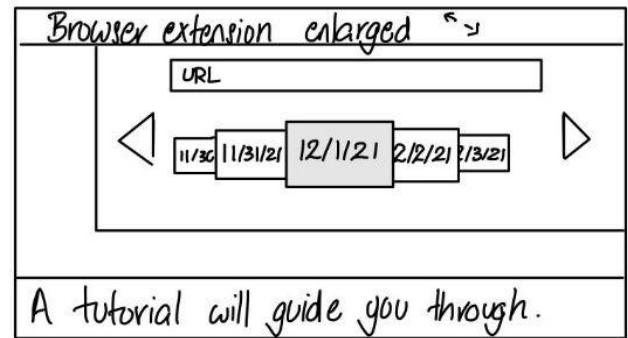
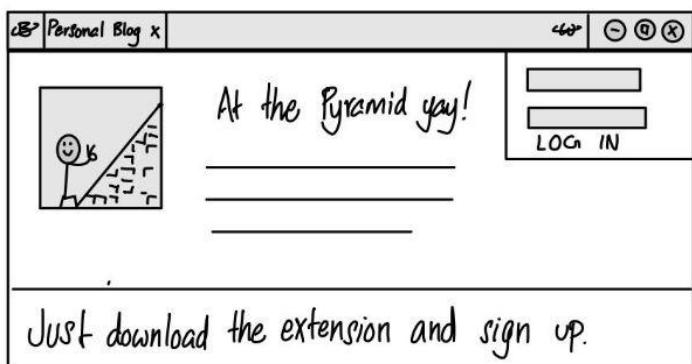
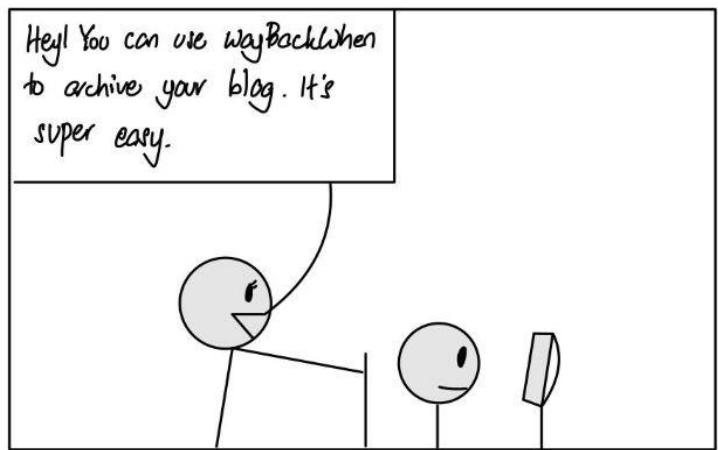
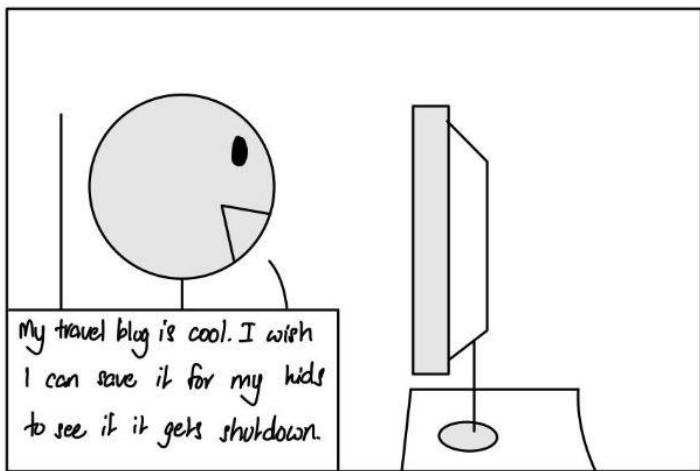


Figure 2: Individual trying to save webpages for memory
Note: The UI for the extension is a placeholder

Storyboard from figure 2 shows the process of a casual user adding their personal webpage or information to save in WayBackWhen. The user this time adds their website through our extensive browser extension. A brief tutorial would guide them through the basic process and functions of the extension. The user is then able to upload their website to our database which would then save updated snapshots of it everyday automatically.

UI Mockup and Prototype

Refer to the Artifact 11 for the link to the Figma file. There the prototype can be utilized and run to visualize functionality.

The Chrome Extension:

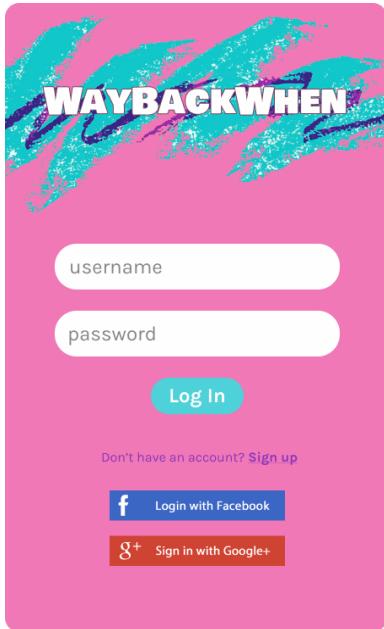


Fig 1. Chrome extension Login

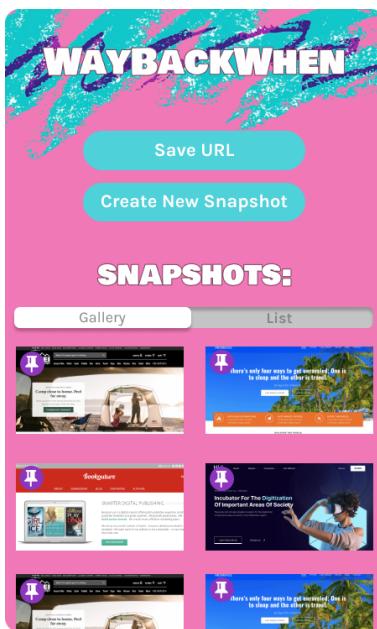


Fig 2. Chrome Extension Landing

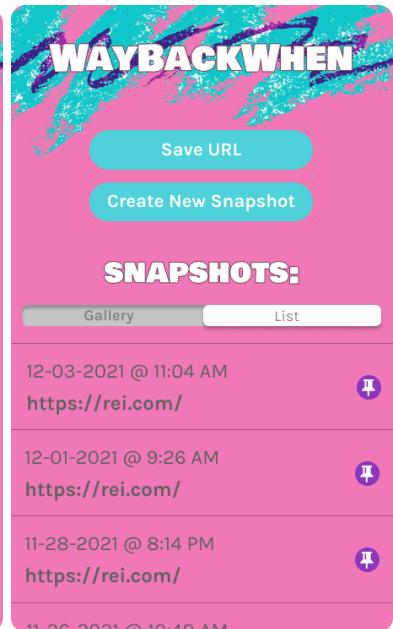


Fig 3. Chrome Extension Landing page 2

The Desktop Version:

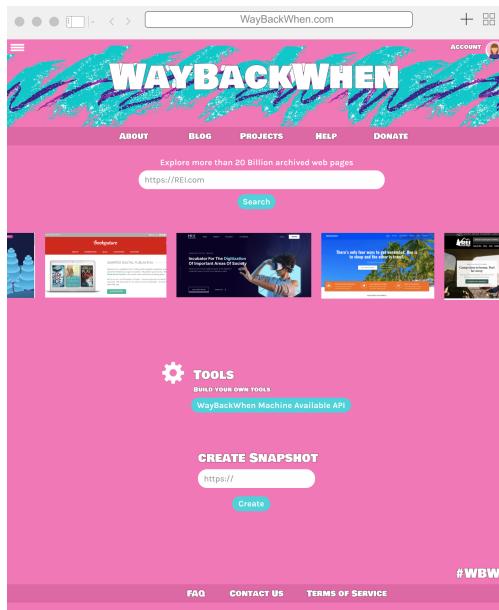


Fig 4. Desktop Home Page

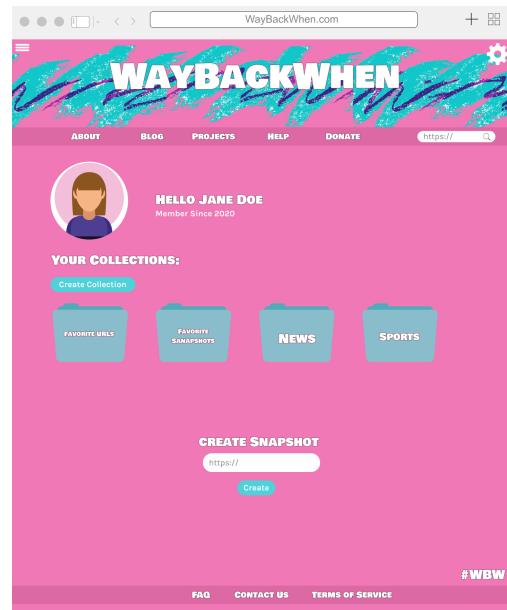


Fig 5. Desktop Account Page

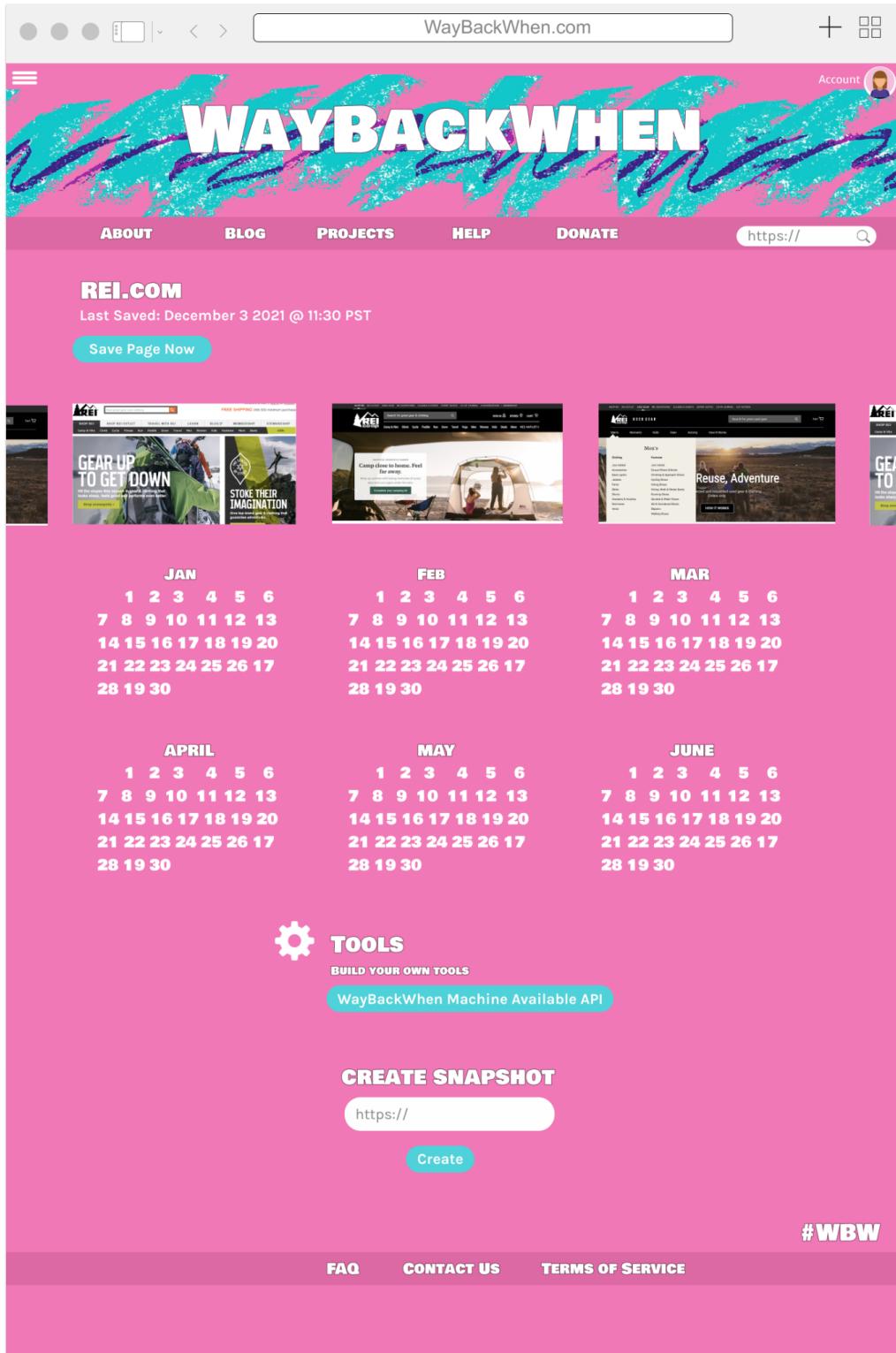


Fig 6. Calendar on desktop version where users can select dates to open past saves

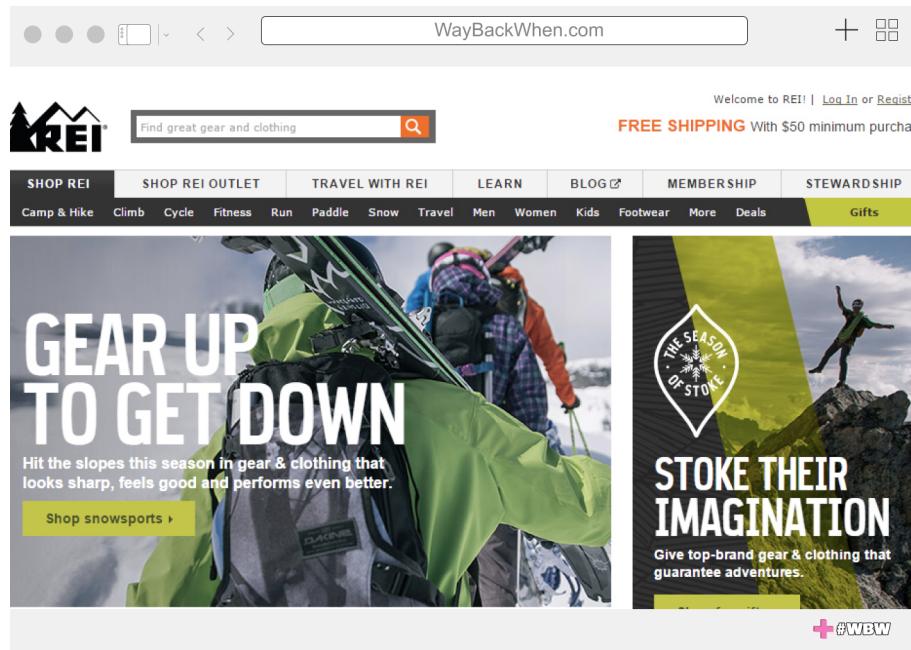


Fig 7. Website that is able to be saved

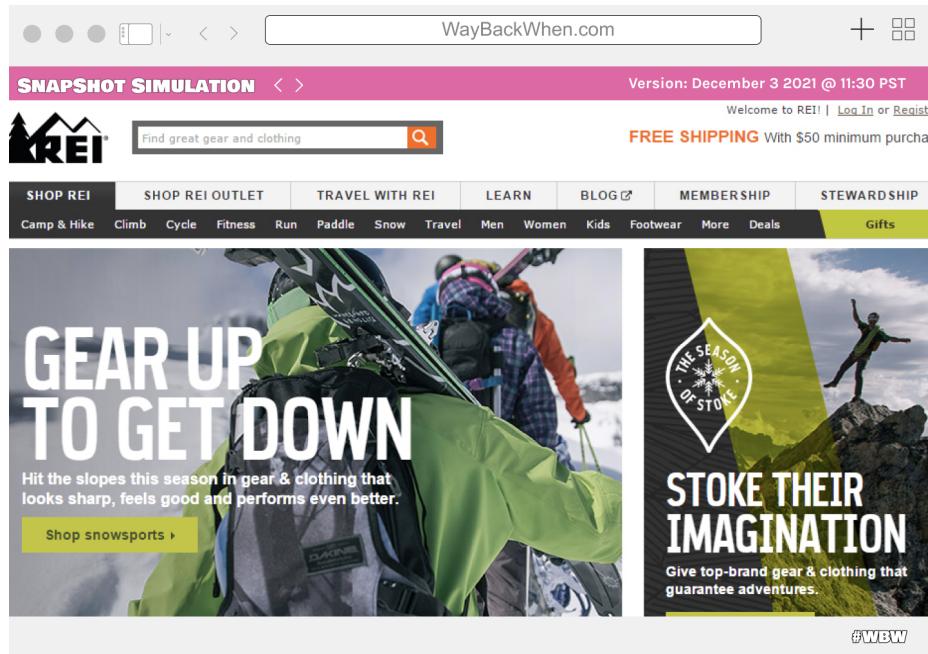


Fig 8. Website that has been saved, being viewed in our website database

Architecture Design

WayBackWhen crawls the web to search for top 2000 publicly accessible web urls and track snapshots of them daily. Websites that the user requests are also tracked per request. Crawling will be outsourced to a third party software for maximum efficiency without requiring more resources. For organizations with restricted data or websites that want to save their websites, Archive-It.org is used as a median to allow them to voluntarily store their websites exclusively on WayBackWhen while individuals who want to store their social medias or websites locked behind an account could just log in to the account to grant us permission. Owners of websites can also request their website to be removed from the crawl list and the archive storage which will block any user from tracking the website and also remove the archives of it. The data will be stored in data centers and will be provided to the users through the cloud. A backup server with encrypted datas will also be set up in case our primary server gets damaged or corrupt. Users of WayBackWhen will have their data stored securely. Passwords for user accounts should all be stored in a hashed format, using SHA-256 to hash and verify passwords.

Pseudocode - website

Search up a web url on website

Take user input as url

If user input is a valid web url in database {

 Show Calendar view on screen

}Else{

 Show error page informing user on non existent web url in database

}

Save or unsave a web url from Calendar view

Get save state of url

If save state is false {

 If user clicks save button {

 Add url to “favorite urls” collections

 }

} else {

 If user clicks unsave button {

 Remove url from “favorite urls” collections

 }

}

Navigate snapshot simulations and save snapshots

if user hovers over a green Calendar day {

 Show snapshot url links

 If user clicks on a snapshot url link {

 Show simulation on whole screen

 If user clicks back button {

```

        If next recent snapshot exists{
            Navigate to next least recent snapshot available
        }
    }
    Else if user clicks forward button {
        If next most snapshot exists{
            Navigate to next most recent snapshot available
        }
    }
    Else if user clicks on a link in snapshot simulation {
        If link is not broken {
            Navigate to link
        }
        Else {
            Show error page
        }
    }
    Else if user clicks x button in top left corner (termination){
        Close simulation
    }
}
}
Else if user clicks on pin next to a snapshot url link {
    Ask user which collection to save to
    If user clicks save button {
        Snapshot is added to user's chosen collection
    }
    Else if user clicks cancel {
        Return to Calendar view
    }
}
Else if user right clicks on snapshot url link {
    Reveal download button
    Prompt user for which file format to save in
    Download file to users computer in corresponding format
}
}

```

View and/or delete saved snapshots and urls

```

If user clicks account button {
    Navigate to account page
    If user right clicks on a collection and collection is not in default collections{
        Reveal delete button
        If user clicks delete button {

```

```
        Delete collection and all of its contents
    }
}

Else if user clicks create new collections {
    Prompt user for collection name
    If user clicks create new collection {
        Create a new collection with collection name
    }
    Else if user clicks cancel {
        Return to account
    }
}

Else if user clicks default url collection "favorite url" {
    Navigate to saved url collection
    If user clicks on url {
        Open a new tab and navigate to the website
    }
    Else if user right clicks on a url {
        Reveal delete button
        If user clicks on delete button {
            Delete url from collection
        }
    }
}

Else if user clicks on a snapshot collection {
    Navigate to the appropriate snapshot collection
    If user clicks on a snapshot image {
        Open a snapshot simulation
        Refer to Navigate snapshot simulations and save snapshots pseudocode within "if user clicks on snapshot url link"
    }
    Else if user right clicks on a snapshot image {
        Reveal delete button
        If user clicks on delete button {
            Remove snapshot from collection
        }
    }
}
```

Create new snapshot of website (at the bottom of any website page)

Take user input in create snapshot url text field

If user clicks create snapshot {

If web url exists {

```

If web url has been saved in last three hours {
    Inform user that url has been saved in last three hours
}
Else {
    Create new snapshot and upload to database
    Prompt user for which collection to save to
    If user clicks save to collection {
        Save snapshot to appropriate collection
    }
    Else if user clicks cancel {
        Return to the current page on website
    }
}
Else {
    Inform user that url does not exist
}
}

```

*Wherever states “click” in the pseudocode represents a left click unless otherwise specified

Pseudocode - browser extension

```

If current web url is not in in database {
    Button becomes a plus button
    If user clicks plus button {
        Create snapshot of website and upload to database
        Prompt user for which collection they would like to save the new snapshot
        If user clicks save to collection {
            Save snapshot to appropriate collection
        }
        Else if user clicks cancel button {
            Do nothing
        }
        Browser extension opens in gallery view
    }
}
Else {
    Button remains a standard button
    If user clicks on button {
        Reveal gallery view of snapshots
        If snapshot of url has not been made in last three hours {
            Create snapshot button is active
        }
    }
}

```

```

Else {
    Create snapshot button is greyed out and inactive
}
If website is in saved urls {
    Save button becomes unsave button
}
Else {
    Save button remains save button
}
if user clicks on snapshot image {
    Open snapshot simulation
    Snapshot fills whole screen
    If user clicks on another snapshot simulation {
        Split screen with first snapshot simulation on the left on the
        second on the right
    }
    Get number of snapshot simulations
    If number of simulations is two {
        For each simulation {
            If user clicks back button {
                If next recent snapshot exists{
                    Navigate to next least recent snapshot
                    available
                }
            }
            Else if user clicks forward button {
                If next most snapshot exists{
                    Navigate to next most recent snapshot
                    available
                }
            }
            Else if user clicks on a link in snapshot simulation {
                If link is not broken {
                    Navigate to link
                }
                Else {
                    Show error page
                }
            }
            Else if user clicks x button in top left corner (termination){
                Close simulation
            }
        }
    }
}

```

```

        Else if number of simulations is one {
            Same interaction as code above under "for each simulation"
        }
    }
    Else if user clicks on pin in top left corner of snapshot image {
        Prompt user for which collection to save to
        If user clicks save to collection {
            Save snapshot to appropriate collection
        }
        Else if user clicks cancel {
            Return to gallery view on browser extension
        }
    }
    Else if user clicks on create snapshot button and create snapshot button is active {
        Snapshot is created and uploaded to database
        Prompt user for which collection to save to
        If user clicks save to collection {
            Save snapshot to appropriate collection
        }
        Else if user clicks cancel {
            Return to current page on browser extension
        }
    }
    Else if user clicks save button {
        If save button is save button {
            Save url to default collection "favorite urls"
        }
        Else {
            Remove url from default collection "favorite urls"
        }
    }
    Else if user right clicks on snapshot image {
        Reveal download button
        Prompt user for which file format to save in
        If user clicks download {
            Download file to users computer in corresponding format
        }
        Else if user clicks cancel {
            Return to gallery view on browser extension
        }
    }
    Else if user clicks on list button {
        Reveal list view of snapshots

```

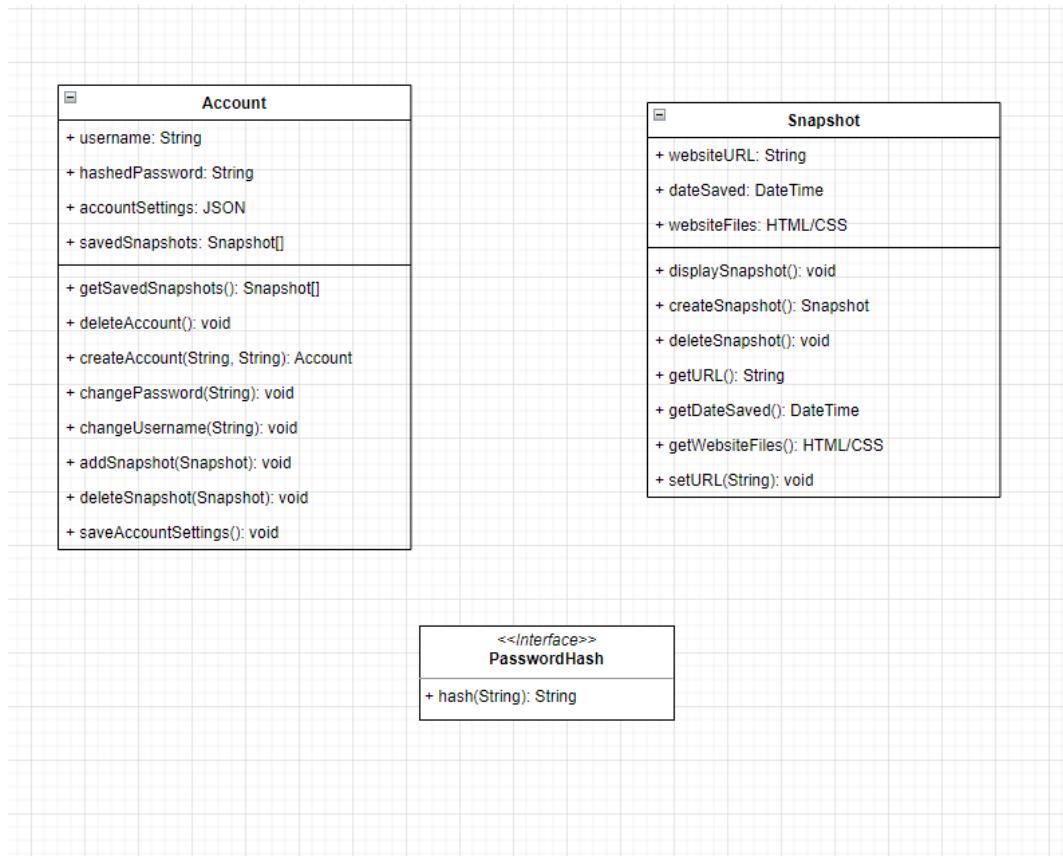
```

        }
    }
}

```

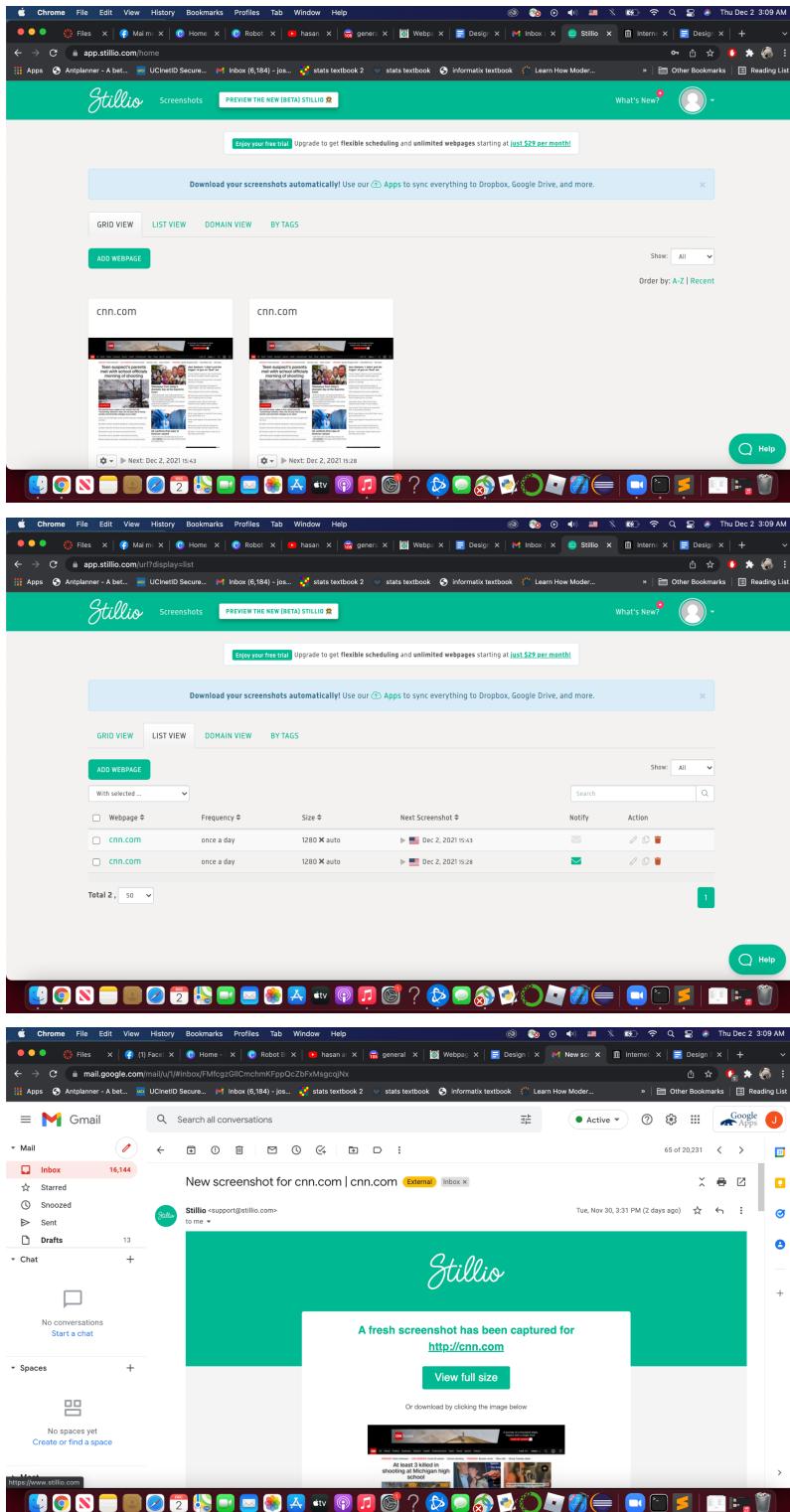
*Wherever states “click” in the pseudocode represents a left click unless otherwise specified

UML



Appendix

- Artifact 1:
 - [!\[\]\(07929003c057e55bca40fcc7e42910ee_img.jpg\) Design Studio 2 11/29 Artifact](#)
 - Students present: Joshua, Htoo, Ian
 - Joshua worked on Feature Comparison
 - Joshua, Htoo, and Ian worked on generating ideas
 - Htoo worked on the Mind Map
- Artifact 2:
 - [!\[\]\(8d9a6952b9fb8de2104cbe836d43bc80_img.jpg\) Design Studio 3 11/30 Artifact](#)
 - Students present: Joshua, Mairie, Claire
 - Joshua worked on Feature Comparison, and Application Design (Audience, Stakeholders, Goals, Decisions, Ideas)
 - Joshua, Claire, Mairie decided on how to split up work
- Artifact 3:
 - [!\[\]\(661fefe957c8a20214f8f1b6e8d0d087_img.jpg\) Design Studio 3 12/1 Artifact](#)
 - Students present: Joshua, Mairie, Claire, Htoo, Ian
 - Joshua, worked on Application Design (Goals)
 - Joshua worked on Interaction Design (Goals, Decisions, Ideas)
 - Claire worked on Application Design (Constraints, Assumptions)
 - Ian worked on UML Diagram
 - Mairie worked on Personas
- Artifact 4:
 - [!\[\]\(b6f1e6d90aef4fdb8e7a844386772852_img.jpg\) Design Studio Artifact 12-2](#)
 - Students present: Joshua, Mairie, Claire, Htoo
 - Joshua worked on Interaction Design (Goals, Constraints, Assumptions, Decisions, Ideas)
 - Joshua worked on Pseudocode
 - Htoo worked on storyboards and Overview
 - Mairie worked on Personas
 - Claire worked on prototype
- Artifact 5 (this document):
 - Students present: Joshua, Mairie, Claire, Htoo, Ian
 - Joshua worked on updating Application and Interaction Design Goals and updated Pseudocode
 - Claire and Mairie worked on UI mockups
 - Htoo worked on Application Design summary and Architecture Design overview
- Artifact 6
 - [WayBack Machine](#)
- Artifact 7
 - [archive.today](#)
- Artifact 8
 - [stillio](#)



- Artifact 9: UML Artifact
 - <https://drive.google.com/file/d/17ju2jHy6R1UWk9lokrOIFD65Qpx39o2Q/view?usp=sharing>

- Artifact 10: Mind map
- Artifact 11: UI Design
 - [Link to Figma prototype](#)