Design Use Cases

Project Manager	Yitian Wang
Business Analyst	Prem Shelat
Senior System Analyst	Satyam Gupta
Software Architect	Wenxiao Li
Software Development Lead	Yuliang Cai
Algorithm Specialist	Haihao Sun
Database Specialist	Fei Dai, Chen Yang
Quality Assurance Lead	Bethany Arellano
User Interface Specialist	Zhirong Lin, Xinyue Zheng

Content:

Account (A)

Social (S)

Starting Session(SS)

Sunflower Core(SF)

Pause (P)

Design Use Case Priority Level Legend	
Must Have	These functionalities are the most basic features of sunflower application, including functions about session and planting sunflowers of the app. The whole app will fail without those functions
Should Have	These functionalities are highly recommended for a comprehensive application, generally including social and account features of the app. Those functions should bring better user experience.
Could Have	These functions are nice to have, but the lack of those functionalities will not affect the user experience.

C: Cancelled
NS: Not Started
WIP: Work In Progress
F: Finished

Account-1 Sign Up

1 - A1, Sign Up

Description	To keep track of users' previous studying statistics and personalized allowlist/blocklist, the system requires the user to create an account prior. The users will be able to create an account by registering.
Desired Outcome	An account will be created for the users from the specified username, password, and email. Users will be able to log into this account.
User Goals	Users shall have an account to participate in the chrome extension's activities.
Primary Actor	User
Dependency Use Cases	None
Priority Level	Must Have
Status	F: Finished
Pre-conditions	None
Post-conditions	The user will have an account registered with the system.
Trigger	The user wants to create an account.
Workflow	1. The user shall click extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon 5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall not be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the sign in with email 9. ProfilePage will call signin.js to config for firebaseui page 10. signin.js shall render firebaseui in `'#firebaseui-auth-container'` 11. The system shall validate the input provided by the user 12. The system shall populate the corresponding fields in the database as the user creates the account.

	13. The user shall be on the HomePage
Alternate Workflow	Cancel signing in: 1. The user shall cancel sign-in process 2. The system shall return Already signed in: 1. The user shall be already signed in 2. The system shall render the User.

Account-2 Login

2 - A2, Login with email

Description	To keep track of users' previous studying statistics and personalized allowlist/blocklist, the system requires the user to create an account prior. The user will be able to create an account by registering.
Desired Outcome	An account will be created for the user from the specified username, password, and email. The user will be able to log into this account.
User Goals	The user shall have an account to participate in the chrome extension's activities.
Primary Actor	User
Dependency Use Cases	A1
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user has an account registered with the system.
Post-conditions	The user has logged in to the application.
Trigger	The user wants to log into their account.
Workflow	1. The user shall click extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon

	5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall not be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the sign in button 9. ProfilePage will call signin.js to config for firebaseui page 10. signin.js shall render firebaseui in `'#firebaseui-auth-container'` 11. The user shall be logged in 12. The user shall be on the HomePage
Alternate Workflow	1. The user enters the wrong email or password in the place to enter their credentials. 2. The system shall produce an error that the credentials entered are wrong. 3. The user will be prompted to enter in the correct credentials to access their account.

Account-3 Logout

3 - A3, Logout

Description	To keep the user's information safe, the user can log out of their account after finishing using the application.
Desired Outcome	The user will be logged out of their account after clicking the logout button.
User Goals	The user shall log out of their account so the user's information can not be accessed.
Primary Actor	User
Dependency Use Cases	A1, A2
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user is logged in to the application and is on the Profile page of the extension.
Post-conditions	The user has logged out of their account.

Trigger	The user wants to log out of their account.
Workflow	1. The user shall click extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon 5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the sign out button at the bottom of the profile page 9. The system shall sign the user out such that the information of the user is safe.
Alternate Workflow	N/A

Account-4 Forgot Password

4 - A4, Forgot Password

Description	The user has forgotten their password and wishes to recover it from the system. The user can recover their password by providing their email to the system.
Desired Outcome	The user receives the password for their account at the email specified.
User Goals	The user wants to obtain their password for their account.
Primary Actor	User
Dependency Use Cases	A1
Priority Level	Must Have
Status	F: Finished

Pre-conditions	None
Post-conditions	The user recovers their password and can now log into the application.
Trigger	The user wants to recover their password.
Workflow	1. The user shall click extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon 5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall not be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the forget password button at the bottom of the profile page 9. The system shall send an email regarding the link which helps the user reset the password
Alternate Workflow	1. The user enters the wrong email in the place to enter their email. 2. The system shall produce an error that the email entered is not associated with a current account. 3. The user will be prompted to enter in the correct email to their account.

Account-5 Login with Google Account

5 - A5, Login with Google Account

Description	To keep track of users' previous studying statistics and personalized allowlist/blocklist, the system requires the user to create an account prior. The user will be able to create an account by registering.
Desired Outcome	The user shall be able to provide their Google email account to access the Sunflower extension.
User Goals	The user wants to access their account information and use the application.
Primary Actor	User
Dependency Use	A1

Cases	
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user has a registered Google account but is not logged in.
Post-conditions	The user can view their study history, modify the allowlist/blocklist, and start the study session
Trigger	The user wants to use the application and its core features.
Workflow	1. The user shall click extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon 5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall not be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the sign in with google button at the bottom of the profile page 9. The system shall populate the user accounts' information in the database such that it is valid for future activity such as start study session, add allowlist, ect
Alternate Workflow	

Account-6 Change Password

6 - A6, Change Password

Description	The user shall be able to change the password for their account in the account settings.
Desired Outcome	The user's password shall be changed and the User shall be able to log in with the new password in the future.
User Goals	The User wants to use a different password in the future.
Primary Actor	User
Dependency Use Cases	A1, A2

Priority Level	**Could Have**
Status	C: Cancelled
Pre-conditions	1. The user has registered an account. 2. The user has logged in. 3. The user knows their current password. 4. The user has navigated to the Settings page.
Post-conditions	1. The user's password is changed. 2. The user can log in with the new password.
Trigger	The user wants to change their password.
Workflow	1. The user shall click the extension icon 2. Popup.jsx shall render the Popup window with default page set to HomePage rendered by HomePage.jsx 3. The user shall be on the HomePage 4. The user shall click the profile icon 5. ProfilePage.jsx shall check whether the user is signed in 6. The user shall not be signed in 7. ProfilePage.jsx shall render sign in page 8. User shall click the forget password button at the bottom of the profile page 9. The system shall send an email regarding the link which helps the user reset the password
Alternate Workflow	

Social-1a Send Friend Request

7 - S1a, Send Friend Request

Description	The user shall send a friend request to another user.
Desired Outcome	The user shall send a friend request to another user they want to add friends with.

User Goals	The user wants to add friends so they can see their sunflowers and react with them on the leaderboard.
Primary Actor	User
Dependency Use Cases	A2, S2, S3
Priority Level	Should Have
Status	F: Finished
Pre-conditions	The user didn't have someone as a friend that they want to add as a friend.
Post-conditions	The user sends a friend request to another user they want to add friends with, and the request is received on the other end.
Trigger	The user shall add friends to see and react to their progress.
Workflow	1. The socialPage.jsx shall render a friend page. 2. The user shall click the add friend button. 3. The user shall search friend's email and click send request. 4. Friend.jsx shall evoke addFriendAction **controller** and call friendRequestHandle in friend.js. 5. Friend.js shall send the message to router.js and a router listens to that message. 6. The router delivers the message to friendDAOManager.js. 7. friendDAOManager.js decides which DAO **model** to call. 8. The DAO **model** interacts with Firebase to send a request to another user whose email matches the user's input. 9. The user shall wait for the receiver to accept/reject the request.
Alternate Workflow	1. The user enters a friend name that doesn't exist in the database. 2. The system shall detect that the friend name they are searching for does not exist, and deny the request to be sent. 1. Another user receiving the friend request does not accept the friend request. 2. The user shall not be able to add them as a friend, and the request shall be gone from the receiver's notifications.

Social-1b Accept Friend Request

7 - S1b, Accept Friend Request

Description	The user shall accept the friend request sent by another user.
Desired Outcome	The user shall accept a friend request. The request sender and the user become mutual friends.
User Goals	The user wants to add friends so they can see their sunflowers and react with them on the leaderboard.
Primary Actor	User
Dependency Use Cases	A2, S2, S3, S1a
Priority Level	Should Have
Status	F: Finished
Pre-conditions	The user received a friend request. The user didn't have the request sender as a friend.
Post-conditions	The request sender and the user become mutual friends.
Trigger	The user shall add friends to see and react to their progress.
Workflow	1. The user shall go to the notification board page and **view** their friend requests. 2. The user shall click the accept request button. 3. Notifications.jsx shall evoke acceptFriendAction **controller** and call acceptFriendRequestHandle in notif.js. 4. Notif.js shall send the message to router.js and a router listens to that message. 5. The router delivers the message to friendDAOManager.js. 6. friendDAOManager.js decides which DAO **model** to call. 7. The DAO **model** interacts with Firebase to terminate the request, and make the request sender and the user mutual friends.
Alternate Workflow	N/A

Social-1c Reject Friend Request

7 - S1c, Reject Friend Request

Description	The user shall reject the friend request sent by another user.
Desired Outcome	The user shall reject a friend request. The request sender and the user will not become friends.
User Goals	The user wants to reject the friend request so they will not become friends with the request sender.
Primary Actor	User
Dependency Use Cases	A2, S2, S3, S1a
Priority Level	Should Have
Status	F: Finished
Pre-conditions	The user received a friend request. The user didn't have the request sender as a friend.
Post-conditions	The request sender and the user will not become friends.
Trigger	The user does not want to add friends with the request sender.
Workflow	1. The user shall go to the notification board page and **view** their friend requests. 2. The user shall click the reject request button. 3. Notifications.jsx shall evoke the rejectFriendAction **controller** and call deleteFriendRequestHandle in notif.js. 4. Notif.js shall send the message to router.js and a router listens to that message. 5. The router delivers the message to friendDAOManager.js. 6. friendDAOManager.js decides which DAO **model** to call. 7. The DAO **model** interacts with Firebase to terminate the request.
Alternate Workflow	N/A

Social-2 View Leaderboard

8 - S2, View Leaderboard

Description	To enhance working efficiency and help users gain satisfaction, the users are able to view the leaderboard which ranks the number of sunflower obtained by all their friends in our application
Desired Outcome	The users are able to view the leaderboard which ranks number of sunflowers owned by all their friends in the database
User Goals	The users want to know their relative ranking among their friends and gain satisfaction.
Primary Actor	User
Dependency Use Cases	A1, A3, SF6
Priority Level	Should Have
Status	F:Finished
Pre-conditions	The users shall successfully log into the system, stays on the social page
Post-conditions	The users will be able to see their leaderboards
Trigger	1. The users click the social page button to view their leaderboards 2. The users click the leaderboard button to view their leaderboards(when the users are already on the social page)
Workflow	1. The users shall login and go to the social page. 2. The Socialpage.jsx in **view** shall show the component Leaderboard.jsx. 3. The Leaderboard.jsx in **view** shall send a message to **controller** file leaderboardRoutes.js . 4. The **controller** files shall trigger **model** file leaderboardlistene.js to fetch real-time data of leaderboard. 5. The **model** file shall send real-time data back to Leaderboard.jsx in **view** 6. The Leaderboard.jsx in **view** shall display the

	update-to-date leaderboard 7. The users shall be able to view their friends' sunflower numbers with reactions rendered on the side
Alternate Workflow	1. The users shall login and be on other sections of socialpage(e.g Friends or Notifications) 2. The users shall click on leaderboard button 3. The Leaderboard.jsx in **view** shall send a message to **controller** file leaderboardRoutes.js . 4. The **controller** files shall trigger **model** file leaderboardlistener.js to fetch real-time data of leaderboard. 5. The **model** leaderboardlistener.js shall send real-time data back to Leaderboard.jsx in **view** 6. The Leaderboard.jsx in **view** shall display the update-to-date leaderboard 7. The users shall be able to view their friends' sunflower numbers with reactions rendered on the side

Social-3 Leaderboard Reaction

9 - S3, Leaderboard Reaction

Description	The users shall be able to add emojis to their friends' sunflower numbers
Desired Outcome	The users shall be able to see their reactions on their friends' sunflower numbers and their friends' reactions on their sunflower numbers
User Goals	The users want to add reactions to their friends' ranks
Primary Actor	User
Dependency Use Cases	A1, A2, S1, S2
Priority Level	**Could Have**
Status	F:Finished
Pre-conditions	The users have friends The users want to react to friends' sunflower numbers

Post-conditions	The users successfully add emojis to their friends' sunflower numbers The users shall be able to see their friends reactions on their sunflower numbers
Trigger	The user wants to see their friends on the leaderboard The user wants to react to their friends' ranks
Workflow	1. The users shall login and go to the social page. 2. The Socialpage.jsx in **view** shall show the component Leaderboard.jsx. 3. The Leaderboard.jsx in **view** shall render the update-to-date leaderboard based on the fetched data. 4. The users shall click on the emoji buttons 5. The Leaderboard.js shall send message to **controller** file reactionRoutes.js 6. The reactionRoutes.js in **controller** shall trigger reactionAction.js and reactionDAOManager.js in **controller** to call reactionDAO.js in **model** 7. The reactionDAO in **model** shall update the latest emojis to database 8. The Leaderboard.jsx in **view** shall render the updated leaderboard with latest emojis rendered on the side.
Alternate Workflow	1. The users shall login and go to the social page. 2. The Socialpage.jsx in **view** shall show the component Leaderboard.jsx. 3. The Leaderboard.jsx shall render the update-to-date leaderboard based on the fetched data. 4. The users shall be able to view their friends' sunflower numbers with reactions rendered on the side 5. If the user has not liked this friend's emoji before, the user shall be able to click the reactions emojis again 6. The updateReactions.js shall decrement the counter of likes for that emojis 7. The users shall be able to see that the number of reaction likes decremented

Social-4 Remove Friends

10 - S4, Remove Friends

Description	The user removes a friend. The friend they remove also loses friend connection to them.
Desired Outcome	The user removes a friend. The friend they remove also loses friend connection to them.
User Goals	The user wants to remove a friend.
Primary Actor	User
Dependency Use Cases	A3, S1a, S1b
Priority Level	Should Have
Status	F: Finished
Pre-conditions	The user has a friend they want to remove.
Post-conditions	The user's friend list is modified and they no longer see the friend removed on their friend list.
Trigger	The user wants to remove a friend from their friend list.
Workflow	1. The SocialPage.jsx shall render a friend page. 2. The user shall click the delete friend button. 3. Friend.jsx shall evoke deleteFriendAction **controller** and call deleteFriendHandle and deleteFriendMutualHandle in friend.js. 4. Friend.js shall send messages to router.js and a router listens to that message. 5. The router delivers the messages to friendDAOManager.js. 6. friendDAOManager.js decides which DAO **model** to call. 7. The DAO **model** interacts with Firebase to remove the friend from their friend list, and remove the user from the friend list of the removed friend.
Alternate Workflow	N/A

Starting Session-1 Add “allowlist” Websites

11 - SS1, Add “allowlist” Websites

Description	To add websites to the “allowlist”, the user is able to add those websites which are safe to visit to during the “allowlist mode”
Desired Outcome	The website of user’s choice is added to the “allowlist” and the system marks it as an “allowlist” website.
User Goals	The user wants to select websites which are safe to visit during the session.
Primary Actor	The user
Dependency Use Cases	SS3
Priority Level	Must have
Status	F:Finished
Pre-conditions	The user is on settings page
Post-conditions	The added website shall be in the “allowlist” and marked as “allowlisted”
Trigger	The user clicks the “add website to allowlist” button
Workflow	1. The user shall click the extension icon 2. Chrome extension engine shall display the Popup window 3. The user shall be on Home Page, rendered by HomePage.jsx 4. HomePage.jsx shall render DisplaySession.jsx 5. The user shall click the setting page button 6. The settingPage.jsx in **view** shall render the settings page 7. The user shall click on the “allowlist” button, and click the “refresh” button 8. The settingPage.jsx in **view** shall render the allowlist that the user have 9. The user shall enter the url of the website in the form provided

	10. The setting.js in **controller** shall mark that website as “allowlisted” for future querying 11. The userDAO.js in **model** shall send store the updated allowlist to the firebase
Alternate Workflow	1. The user enters wrong url in the place to enter the url 2. The system shall produce an error that the url entered is wrong. 3. The user enters the url of a website which is already in the list, the system shall not crash if such behavior happened.

Starting Session-2 Add “Blocklist” Websites

12 - SS2, Add “blocklist” Websites

Description	To add websites to the “blocklist”, the user is able to add those websites which should not be visited during the “blocklist mode”
Desired Outcome	The website of user’s choice is added to the “blocklist” and the system marks it as a “blocklisted” website.
User Goals	The user wants to select websites which are not safe to visit during the session
Primary Actor	The user
Dependency Use Cases	SS3
Priority Level	Must have
Status	F:Finished
Pre-conditions	The user is on settings page
Post-conditions	The added website shall be in the “blocklist” and marked as “blocklisted”
Trigger	The user clicks the “add website to blocklist” button
Workflow	1. The user shall click the setting page button 2. The settingPage.jsx in **view** shall render the settings page 3. The user shall click on the “blocklist” button, and click the “refresh” button 4. The settingPage.jsx in **view** shall render the allowlists that the user have 5. The user shall enter the url of the website

	6. The setting.js in **controller** shall mark that website as “blocklist” 7. The userDAO.js in **model** shall send updated blocklist to the firebase for future querying
Alternate Workflow	1. The user enters wrong url in the place to enter the url 2. The system shall produce an error that the url entered is wrong. 3. The user enters the url of a website which is already in the list, the system shall not crash if such behavior happened.

Starting Session-3 Select Mode

13 - SS3, Select Mode

Description	There are two models in our chrome extension, the first one is “allowlist model” where any websites not included in the “allowlist” would be blocked during the study session. Another one is “blocklist model” where any websites include in the “blocklist” would be blocked. In other words, the “blocklist” mode is more tolerant since it only blocks website in the blocklist that the user defined before, and the “allowlist” mode is more strict since it blocks every websites that is not included in the “allowlist”. The model selection requires the user to prepopulate the “allowlist” and “blocklist”
Desired Outcome	User will be able to select their type of focus and be rewarded accordingly
User Goals	The user wants to avoid all the distraction websites or the user wants to be able to visit only some websites.
Primary Actor	User
Dependency Use Cases	S4, SS1
Priority Level	Must Have
Status	F:Finished
Pre-conditions	Users have entered the “allowlist” and “blocklist” websites.
Post-conditions	User is successfully entered into the mode they selected
Trigger	User is filling up the details of the session and wants to customize it further.
Workflow	1. User clicks on Homepage in the popup 2. The HomePage.jsx in **view** shall render the options “allowlist” or “blocklist” mode. 3. Users are asked to choose their type of focus for this session. 4. The session.js in **controller** shall mark the mode as the type chosen, and block the websites based on the selected mode when the study session starts.

Alternate Workflow	N/A

Starting Session-4 Set Time For Session

14 - SS4, Set Time For Session

Description	The user wants to set the study duration for the session.
Desired Outcome	User successfully picks the study duration, and the system will award the corresponding number of sunflowers if the user completes the session successfully.
User Goals	The user wants to set the time for which they are monitored.
Primary Actor	The user
Dependency Use Cases	SS3, SS1, SS2
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user has selected what type of focus they are looking for and now they have to choose for how long they are going to focus.
Post-conditions	The user is now able to start their session.
Trigger	The user has finished selecting their preferences for the type of the session they are looking for.
Workflow	1. User shall login in and stay in the home page, 2. The HomePage.jsx shall render the UI for the study session 3. The user can select the duration of their study session 4. The session.js in **controller** shall record the corresponding duration for the study session, and the
Alternate Workflow	N/A

Starting Session-5 Start Focus Session

15 - SS5, Start Focus Session

Description	The user clicks the start button to begin the study session.
Desired Outcome	The timer for the session shall start, and the system shall starts Blocking the blocked website
User Goals	The user wants to start the focus session.
Primary Actor	User
Dependency Use Cases	SS1, SS2, SS3, SS4
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user has selected allowlist/blocklist mode and also set the focus time.
Post-conditions	The user shall successfully enter the focus session.
Trigger	The user has customized the session and wants to begin the session now.
Workflow	1. The user shall click the extension icon 2. Chrome extension engine shall display the Popup window 3. The user shall be on Home Page, rendered by HomePage.jsx 4. HomePage.jsx shall render DisplaySession.jsx 5. The user shall click the start button from the DisplaySession.jsx 6. DisplaySession.jsx shall send ‘start-session’ to the background to start the session. 7. sessionRoute.js shall correspond ‘start-session’ to startSessionAction in the sessionAction.js on receiving messages. 8. startSessionAction shall start the session. 9. session.js should start the timer. 10. session.js shall record the Date-time of this moment 11. session.js shall inject scripts to all current tabs and start

	listening for new or reloaded tabs and inject scripts to them. 12. session.js shall send a message to DisplaySession.jsx on the HomePage.jsx every second. 13. The DisplaySession.jsx shall display the running session view and the remaining time for that session. 14. The overlay.js shall detect whether the website should be blocked.
Alternate Workflow	N/A

Sunflower Core-1 Block websites during the study session

16 - SF1, Block websites during the study session

Description	The user's access to blocked websites will be denied during the session, the system shall inject a layer to the blocked website, and the user will no longer be able to visit the blocked website during the study session.
Desired Outcome	The user will not be able to access the content of the blocked websites
User Goals	The user wants the system to deny their access to the blocked websites during the session
Primary Actor	User
Dependency Use Cases	SS1, SS2, SS3, SS4, SS5
Priority Level	Must Have
Status	F: Finished
Pre-conditions	The user has selected mode. The user has selected allowlist websites and blocklist websites. The user has set the length of the session. The user has started the session.
Post-conditions	1. The user resumes the session. 2. The user quits the session 3. The user pauses the session 4. The countdown ends and the user completes the session
Trigger	The user wants to avoid temptations from non-academic websites and stay focused.
Workflow	**Main Workflow: blocklist mode and website that should be blocked** 1. The user shall be in session. **Background:** 2. session.js shall call overlay.js and send messages about

	session info to Popup every second during session. 3. overlay.js shall check the session mode. *4. overlay.js shall find the mode is blocklist.* 5. overlay.js shall determine whether each website should be blocked by checking if it shares the same host name as those on the blocklist. 6. overlay.js shall broadcast messages to all currently opened websites to control the Content/index.js. **Content:** *7. Content/index.js shall receive messages indicating the website it injected to should be blocked, and information about the current running session.* 8. Content/index.js shall inject an overlay on top of the website 9. Content/index.js shall disable all keyboard event and mousewheel event 10. Content/index.js shall render DisplaySession.jsx to receive information about current session time and session is currently running. 11. DisplaySession.jsx rendered by Content/index.js shall display the time remaining for this session on the mask layer 12. DisplaySession.jsx rendered by Content/index.js shall display quit button and pause button **Popup:** 13. HomePage.jsx shall render DisplaySession.jsx to receive information about current session time and session is currently running. 14. DisplaySession.jsx rendered by HomePage.jsx shall display the time remaining for this session on the mask layer 15. DisplaySession.jsx rendered by HomePage.jsx shall display quit button and pause button
Alternate Workflow	**Alternative Workflow1: allowlist mode** 1. Perform workflow from 1 to 3 4. *overlay.js shall find the mode is allowlist.* 5. overlay.js shall determine whether each website should be blocked by checking if it does not share the same host name as those on the allowlist. 6. overlay.js shall broadcast messages to all currently opened websites to control the Content/index.js. 7. Go to 7 of Main Workflow or Alternative Workflow 2 for **Content** and **Popup** processing. **Alternative Workflow 2: website that should not be blocked** 1. Perform Main workflow or Alternative Workflow 1 from 1 to 6 **Content:**

	7. *Content/index.js shall receive messages indicating the website it injected to should not be blocked, and information about the current running session.* 8. Content/index.js shall make sure no overlay is injected. 9. Go to 12 of Main Workflow for **Popup** processing.

Sunflower Core-2 View session history

17 - SF2, View session history

Description	User can view their 7-day historical statistics
Desired Outcome	The user can see the user's historical statistics
User Goals	User wants to the user's historical statistics
Primary Actor	User
Dependency Use Cases	A1,A2,SS1,SS2,SS3,SS4,SS5
Priority Level	Should Have
Status	F:Finished
Pre-conditions	The user is logged in and is on a profile page
Post-conditions	The user see the user's historical statistics
Trigger	Users wants to have a record of past activities and progress
Workflow	1. The user shall be on the Home page. 2. The system shall call viewHistory from HomePage.jsx and send a "view_history" message. 3. The system shall receive the message in route.js and call historylistener() function. 4. The system shall retrieve database snapshot and pass it into viewhistory() function in history.js and calculate the Date and focus length of each day as two arrays. 5. The system shall store the chart data into local storage. 6. The user shall be on the social page. 7. The user shall click the Session History. 8. The system shall load chart data from local storage at sessionhistory.jsx and display them as a bar chart in the session history page. 9. The user shall click the back button 10. The user shall go back to the social page
Alternate Workflow	

Sunflower Core-3 Completed the Session

18 - SF3, Completed the Session

Description	The user shall finish the study session. The Sunflowers will be awarded to the user who completed the session.
Desired Outcome	The user finishes the current session which will be recorded and will be able to start a new session.
User Goals	The user has completed the session without quitting.
Primary Actor	User
Dependency Use Cases	S2, SS5, SF2, SF5, SF6
Priority Level	Must Have
Status	F:Finished
Pre-conditions	The user has started a study session with the sunflower planted.
Post-conditions	The session ended and all the Sunflowers have grown 100% (15 mins per one)which could be recorded to the leaderboard.
Trigger	The user finished the session
Workflow	1. The user shall complete the session without quitting 2. The displaySession.jsx session page shall a notification that tells the user about the completion of the session 3. The sessionDAO.js in the background folder shall send the data to firebase, and the number of sunflowers awarded to the user would be recorded in the database. 4. The HomePage.jsx in **view** should render the UI
Alternate Workflow	N/A

Sunflower Core-4 End Study Session Midway

19 - SF4, End Study Session Midway

Description	The user wants to quit the study session, and this will result in having zero sunflowers awarded.
Desired Outcome	The user shall stop the current study session in the middle and start a new session.
User Goals	The user decided to quit the study session.
Primary Actor	User
Dependency Use Cases	SS5
Priority Level	Must Have
Status	F:Finished
Pre-conditions	The user has started a study session with the sunflower planted.
Post-conditions	The user ends the session and loses all the Sunflower. Then the user will be able to start the next session from the Home page.
Trigger	The user wants to quit the session in the middle.
Workflow	1. The user shall click on the “quit” button on the current session page. 2. The system shall prompt the confirmation asking whether the user would like to quit the session. 3. The user shall click the "OK" button. 4. The DisplaySession.js in **view** shall render the message saying that the user failed to complete the session. 5. The user shall click the back button 6. The shall reroute the user to session page again where the user can start another session.
Alternate Workflow	.1. The system asks whether the user wants to end the session, the user clicked cancel 1. The system shall continue counting down the timer

Sunflower Core-5 Check Total Sunflower

20 - SF5, Check Total Sunflower

Description	The user wants to check how many sunflowers planted so far.
Desired Outcome	Able to check number of sunflowers planted
User Goals	The user wants to check their accomplishments or rankings.
Primary Actor	User
Dependency Use Cases	A2, A5, S2
Priority Level	Must Have
Status	F:Finished
Pre-conditions	The user has logged in.
Post-conditions	The user sees how many flowers are planted.
Trigger	The user wants to check their accomplishments or rankings.
Workflow	1. The user shall be on the social page. 2. The system shall import Usercontext from backend user.js in Profile.js. 3. The system shall call setUserSnapshot to get the snapshot from the database. 4. The system shall call getUserSnapshot to send the snapshot to UserContext. 5. The ProfilePage.jsx shall get the total sunflower data from the snapshot and display it in UserProfile.js.
Alternate Workflow	N/A

Pause-1 Pause Session

21 - P1, Pause Session

Description	The user can choose to temporarily pause their study session.
Desired Outcome	Engaging in non-academic activity will not trigger study reminders and not affect their sunflowers.
User Goals	The user can take a break from their study session.
Primary Actor	User
Dependency Use Cases	P2
Priority Level	**Could Have**
Status	F: Finished
Pre-conditions	The user has the pause feature enabled.
Post-conditions	The user's study session is temporarily paused.
Trigger	The user wants to take a study break, but not end their study session altogether.
Workflow	1. The user shall be in the middle of a study session 2. The DisplaySession.jsx in **view** shall present the button labeled "Pause" 3. The user shall click the button labeled "Pause" 4. The session.js in **controller** shall stop monitoring which websites, and increment the number of paused that the user utilized 5. The overlay.js in **controller** shall stop blocking the websites
Alternate Workflow	1. The user shall be in the middle of the study session 2. The DisplaySession.jsx in **view** shall present the button labeled “Pause”

Pause-2 Resume Session

22 - P2, Resume Session

Description	The user will be able to return back to their study session after previously pausing it.
Desired Outcome	The user's online activity will once again be monitored and affect their sunflower growth.
User Goals	The user wants to resume their study session and accessing non-academic websites will affect their sunflowers.
Primary Actor	User
Dependency Use Cases	SS5, P1
Priority Level	**Could Have**
Status	F: Finished
Pre-conditions	The user has started the session
Post-conditions	The user resumes their study session.
Trigger	The user has finished taking their study break and is ready to return to their study session
Workflow	1. The user shall be inw the middle of a study session, and the user shall pause the study session. 2. The DisplaySession.jsx in **view** shall present a button labeled "Resume" 3. The user shall click the button labeled "Resume" 4. The overlay.js in **controller** shall continue monitoring which websites, and in the session page, the DisplaySession.jsx in view shall render the number of remaining pauses that the user could use
Alternate Workflow	N/A