IPO

|  |  |  |
| --- | --- | --- |
| signUp(fullname, password, emailAddr) | | |
| INPUT | PROCESS | OUTPUT |
| **Actions**:   * User clicks sign up   **Parameters:**   * fullname(string) * emailAddr(string) * password   **Files:**   * “salt.csv” * “users.csv” | 1. Open the “salt.csv” file 2. Get the salt from the “salt.csv” file 3. set hashed password to the hashed value of (password+salt) 4. Check if the user already has an associated email signed up. 5. If the user hasn’t already been signed up then write their fullname, hashed password and email to the “users.csv file” | Return True & output users.csv  OR  ReturnFalse(Boolean) & “This email already has an associated account” |

|  |  |  |
| --- | --- | --- |
| loginToApp(email\_input, password\_input): | | |
| INPUT | PROCESS | OUTPUT |
| **Actions:**   * User clicks sign in   **Parameters:**   * email\_input(string) * password\_input(string)   **Global Variables:**   * emailAddr(string) * appPassword(string) * emailPassword(string) | 1. Open “salt.csv” 2. Get the salt from the “salt.csv” file 3. Open “users.csv” 4. Check each row in “users.csv” until the email in the row is equal to email\_input. 5. Set hashedinput to the hashed value of (password\_input + salt) 6. If the password field of the row is equal to hashedinput then set the global variable emailAdrr to email\_input and the global variable appPassword to hashedinput. 7. If the password field doesn’t equal hashedinput then display an error. 8. Prompt the user for their email password and check that it is right until they enter the right password (set the global variable emailPassword to the email they entered into the prompt and return True), or until they close the dialogue (cancel the login attempt and return false.) 9. If there is no row where the email field matches email\_input, display an error message. | “Incorrect password!” and False    True or  False  “This user hasnot signed up” |

|  |  |  |
| --- | --- | --- |
| reload\_wishlists | | |
| INPUT | PROCESS | OUTPUT |
| **Actions:**   * User clicks reload native wishlists   **Global Variables:**   * emailAddr (string) * serverPassword(string) | 1. Connect to gmail server using:    1. ”wishlists.server.gmail.com”,    2. serverPassword    3. server route(“imap.gmail.com”) 2. Search the inbox of “wishlists.servevr@gmail.com” for emails from emailAddr 3. If there are emails from emailAddr then for each email found, download the attachment (the wishlist) | Wishlists downloaded from the emails |

|  |  |  |
| --- | --- | --- |
| reload\_shared\_wishlists | | |
| INPUT | PROCESS | OUTPUT |
| **Actions:**   * User clicks reload external wishlists   **Global Variables:**   * emailAddr (string) * emailPassword(string) | 1. Connect to gmail server using:    1. emailAddr    2. serverPassword    3. server route(“imap.gmail.com”) 2. Search the inbox of emailAddr for emails from “wishlists.servevr@gmail.com” 3. If there are emails from “wishlists.servevr@gmail.com” then for each email found, download the attachment (the wishlist) | Wishlists downloaded from the emails |

|  |  |  |  |
| --- | --- | --- | --- |
| share\_wishlist(to\_addr, wishlist\_filename) | | | |
| INPUT | PROCESS | OUTPUT | |
| **Actions:**   * User clicks share wishlist   **Parameters:**   * to\_addr (string) * wishlist\_filename(string) | 1. Get the email address portion of the wishlist filename. 2. Set sent\_from to the email address portion of the wishlist filename 3. Send an email from wishlists.server@gmail.com’ to to\_addr with the selected wishlist as an attachment | | Email sent to to\_addr with the wishlist attatched |

|  |  |  |  |
| --- | --- | --- | --- |
| save(users\_email, email\_password, wishlist\_filename) | | | |
| INPUT | PROCESS | OUTPUT | |
| **Actions:**   * User clicks save wishlist   **Parameters:**   * users\_email(string) * email\_password * wishlist\_filename | 1. Send an email from users\_email to “wishlists.server@gmail.com” with(wishlist\_filename plus “.csv” at the end of the filename) as an attachment | | Email sent to “wishlists.server@gmail.com” |

|  |  |  |  |
| --- | --- | --- | --- |
| createwishlist(name) | | | |
| INPUT | PROCESS | OUTPUT | |
| **Actions:**   * User clicks create wishlist   **Parameters:**   * Name(string)   **Global Variables:**   * Attatchment-dir | 1. Open a new file called name.csv where name is the parameter entered in attachment-dir 2. Write “item(s)” to the file 3. Close the file | | New wishlist called name.csv with “item(s)” as the first row of text. |

|  |  |  |  |
| --- | --- | --- | --- |
| deletewishlist | | | |
| INPUT | PROCESS | OUTPUT | |
| **Actions:**   * User clicks delete wishlist   **Parameters:**   * Name(string)   **Global Variables:**   * Attatchment-dir | 1. Delete the file name.csv in attatchment dir | | Selected wishlist delted |