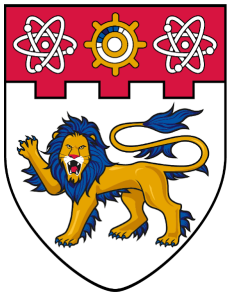


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

School of Computer Science and Engineering (SCSE)

SC20006 Software Engineering

Software Requirement Specification

for Homey

Lab Group/ Group No.: A29/Group 1
Team Name: NGNRS

Team Members:

Crystal Cheong Yu Qing	U2121134A
Chan Ming Han	U2120477J
Ang Jun Koon	U2122362K
Cheong Chi Hian	U2122966L
Ng Mu Rong	U2121323F

1. Product Description

1.1. Purpose

Homey seeks to provide a comprehensive and convenient online platform for homebuyers, sellers, and renters to search for properties, view property listings and connect with agents. The web application is designed to simplify the property search process and make it easy for users to find their dream home or investment property.

With access to a vast database of property listings, Homey offers users a one-stop-shop for all their real estate needs, from finding the right property to connecting with experienced agents who can guide them through the buying, selling or renting process. The goal of Homey is to provide a user-friendly and efficient real estate search experience, helping users to make informed decisions and find the perfect property to suit their needs.

1.2. Scope

The Homey web application offers a range of features and services to facilitate the buying, selling, and renting of properties, including:

(1) Property search: Users can search for properties based on their location, budget, and other criteria, and view detailed information and images of each property.

(2) Property listings: Homey provides access to a vast database of property listings, complete with detailed information, high-quality images, and pricing information.

(3) Agent connection: Users can connect with experienced agents who can provide guidance and assistance throughout the buying, selling, or renting process.

(4) User account management: Users can create and manage their accounts, save property searches, and receive updates on new listings and properties that meet their criteria.

1.3. Users and Stakeholders

Users

- Homebuyers
- Home sellers
- Renters
- Real estate investors
- Real estate agents

Stakeholders

- Homey development team
- Homey management team
- Real estate agents and agencies partnering with Homey
- Potential investors in Homey
- Advertisers using Homey to promote properties or services.

1.4. Assumptions and Constraints

Assumptions

Users should have a device that is able to connect to the internet

Constraints

The web application is only available in English, language location support may be available in the future iterations

1.5. Initial UI Mockups

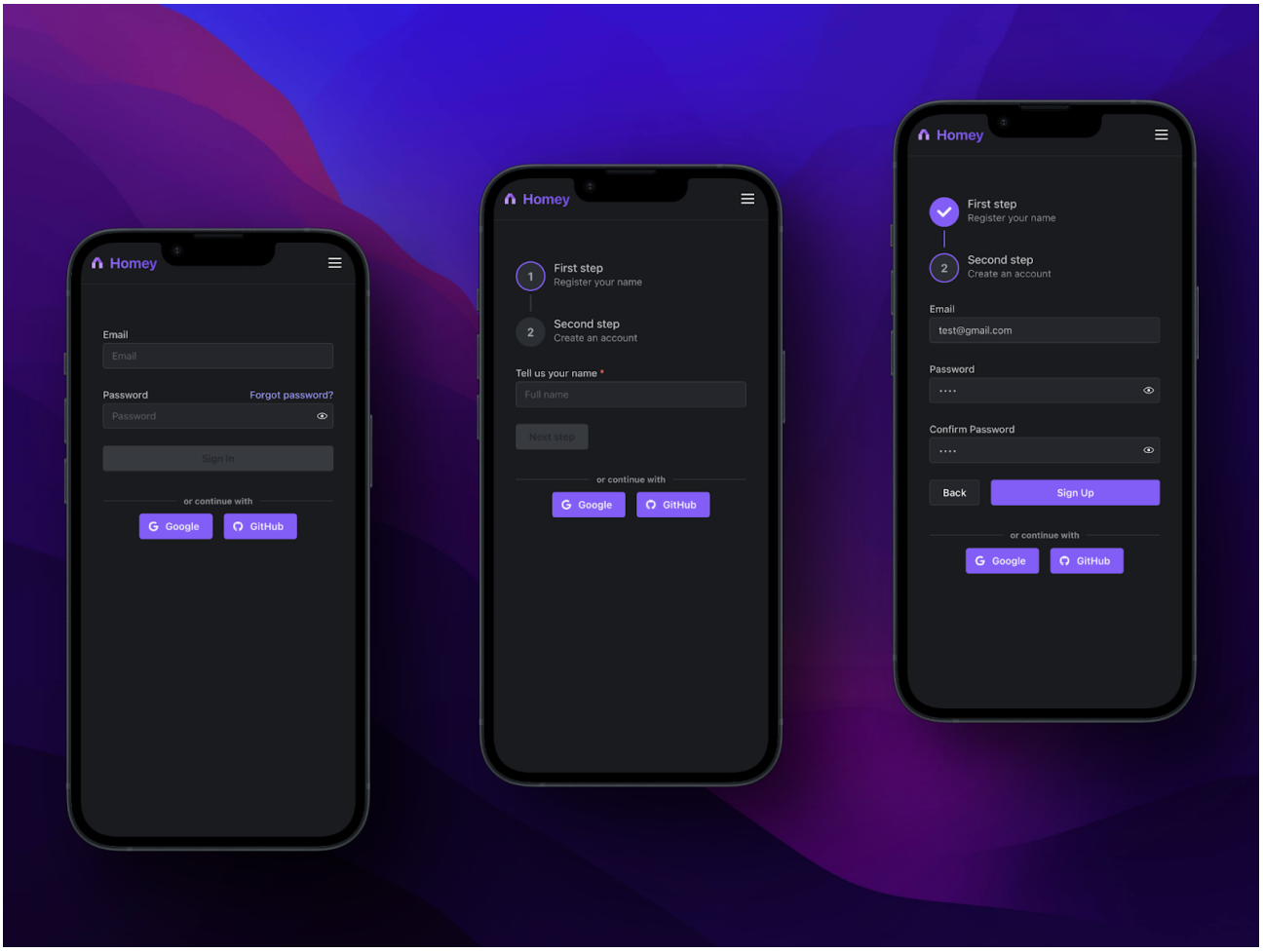


Figure 1.5.1, Sign In and Sign Up pages (mobile 320px)

Figure 1.5.1 shows the mobile mockups of the Sign In and Sign Up screens. The design is consistent with the overall aesthetic of the application, featuring the brand colour scheme and intuitive layout. Both screens offer the option for the user to authenticate via any one of the supported external providers.

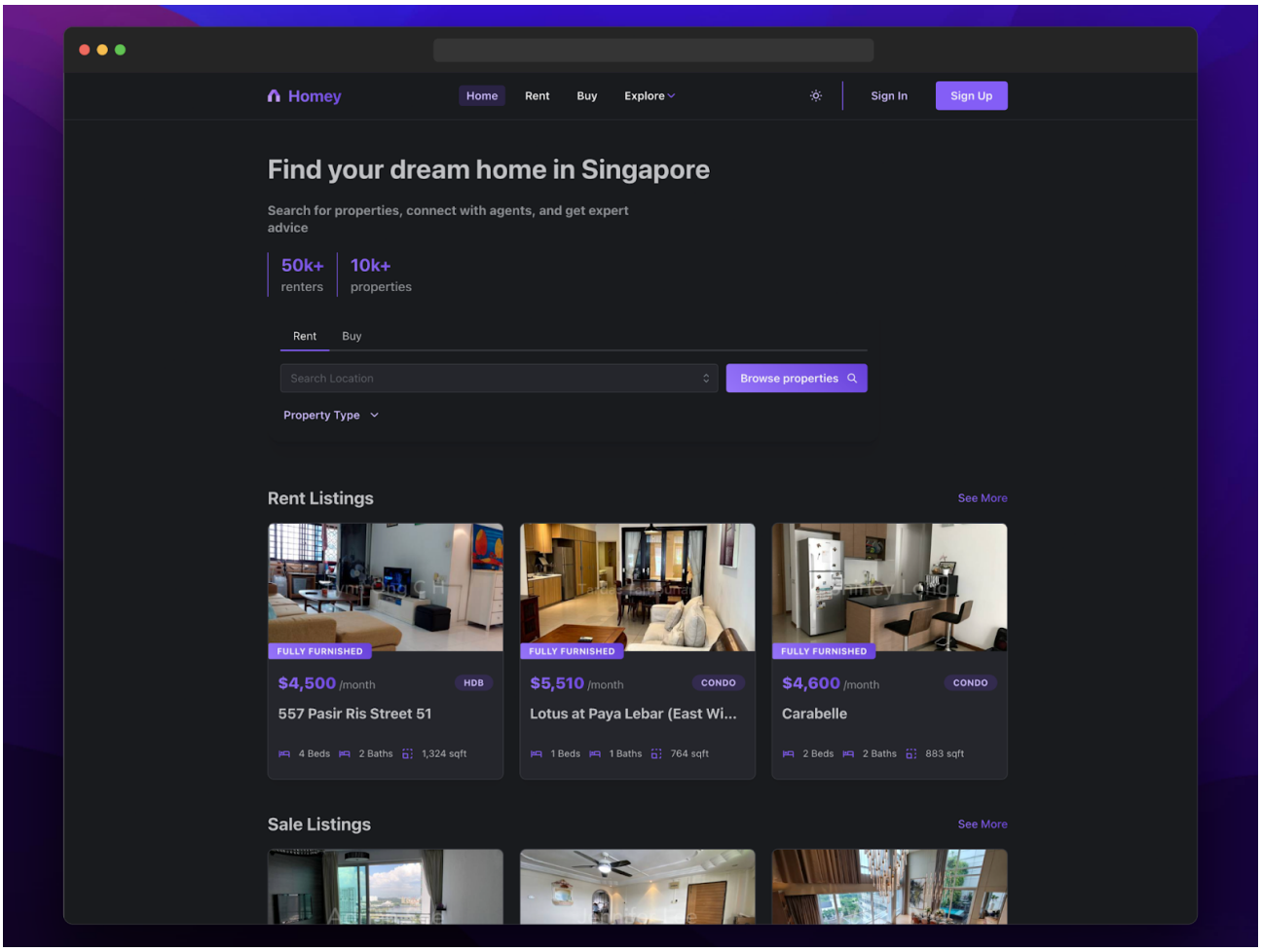


Figure 1.5.2, Landing page (desktop 1024px)

Figure 1.5.2 shows the landing page of Homey which features a large hero section prominently displayed at the top of the page. This section includes a multiselect location input field and a dropdown with property types to search listings. Below the hero section, the page showcases some of the featured listings currently available for either rent or sale, along with eye-catching photos and a brief description of each property.

2. Functional Requirements

2.1. Account

Index / Functionality	Requirements
2.1.1 Sign Up	1. The user should be able to register for a new account 1.1. The required information includes: 1.1.1. Name 1.1.2. Email Address 1.1.3. Password & Confirm Password 1.2. The system should validate that all required fields have the sufficient inputs. 1.3. The system should validate that the input email address is unique and not part of an existing account. 1.3.1. If the input email address is not unique, system should display an error message 1.4. The system should validate that Password and Confirm Password matches. 1.4.1. If either input password or confirm password does not match, the system should display an error message. 1.5. The system should use the Argon2 hashing algorithm to perform salt-hashing on the input password before storing into the database. 2. Alternatively, the user should be able to register for a new account via supported external provider(s) 2.1. Supported externals providers includes: 2.1.1. Google 3. Upon successful sign up, the system should automatically sign in with the newly created account. 4. The system should create a new Session token consisting of the account information 5. The system should redirect the user to the landing page.
2.1.2 Sign In	1. The user should be able to sign in with their email and password 1.1. The system should validate that all required fields have the sufficient inputs. 1.2. The system should validate that the input email address is part of an existing account 1.3. The system will perform reverse salt-hashing on the database-stored password to compare with the input password.

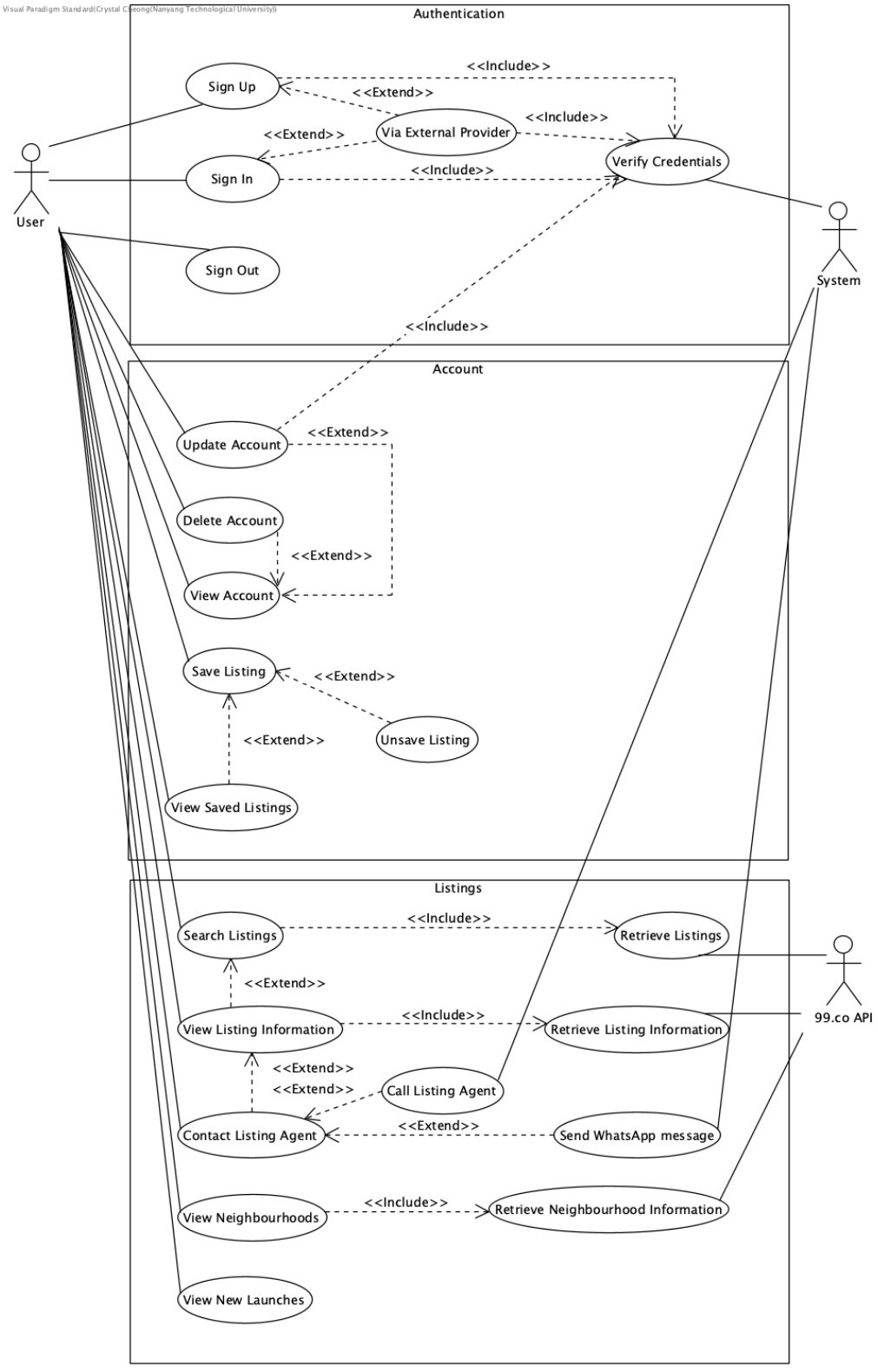
	2. Alternatively, the user should be able to sign in via support external provider(s). 2.1. Supported externals providers includes: 2.1.1. Google 3. Upon successful sign in, the system should create a new Session token consisting of the account information 4. The system should redirect the user to the landing page.
2.1.3 Sign Out	1. The user should be able to sign out 2. The system should destroy the current Session token associated with the user's account 3. Upon successful sign out, the system should display `Sign In` and `Sign Up` buttons in place of the account information in the navigation bar.
2.1.3 Update Account Information	1. The user should be able to view their account information 2. The user should be able to edit their account information 2.1. Editable fields includes: 2.1.1. Name 2.2. The system should validate that at least one of the editable fields has input value 2.3. The system should validate that the input value differs from the current account information 2.4. Upon successful validation, the user should be able to click on the `Save Changes` button to submit their changes 2.5. The system should perform an update operation on the database record with the matching account email address to update the account information 2.6. If unsuccessful, the system should display an error message. 2.7. The system should display the updated account information.
2.1.4 Delete Account	1. The user should be able to request for account deletion 2. Upon the user clicking the `Delete Account` button, the system will perform a delete operation on the database record with the matching account email address. 3. If unsuccessful, the system should display an error message. 4. The system will destroy the current Session token associated with the deleted account.
2.1.5 Save Listing	1. The user must log in first in order to save their favourite listing 2. The user should be able to save their favourite listing 3. The user should be able to view all their saved listings

	4. The user should be able to unsave specific saved listings

2.2. Listings

Index / Functionality	Requirements
2.2.1 Search	1. The user should be able to search for property listings based on filters 1.1. Filters includes: 1.1.1. Location 1.1.2. Property Type 1.1.2.1. HDB 1.1.2.2. Condo 1.1.2.3. Landed 1.2. The system should retrieve the location input from a multiselect field that allows at most 3 locations. 1.3. The system should provide autocompletion for the location multiselect input field 1.4. The system should be able to query for property listings based on the given filters 1.5. If no listings are available based on the filters, the system display an error message
2.2.2 View Listing Information	1. The system should be able to query for the specific listing with the route parameters. 1.1. Route parameters includes: 1.1.1. Listing ID 1.1.2. Cluster ID 2. The system should be able to retrieve the specific listing information from the 99.co API 3. The system should be able to display information about the specific listing. 3.1. Information includes: 3.1.1. Address Name 3.1.2. Listing Photos 3.1.3. Listing Description 3.1.4. Listing Agent Information 3.1.5. Listing Agent Contract Information 3.1.6. Embedded Google Map view of the listing location 3.2. The system should display a `Save` button for authenticated users to save the listing

3. Use Case Diagram



4. Use Case Descriptions

4.1. Account

Use Case ID:	11		
Use Case Name:	Sign In		
Created By:	Crystal Cheong	Last Updated By:	
Date Created:	29/01/2023	Date Last Updated:	

Actor:	User, System
Description:	This use case describes the sign-in authentication process.
Preconditions:	- Device must be connected to the internet - User account must already exist in the database
Postconditions:	- User is able to save listings - User is able to view account information - Account name is shown in the navbar
Priority:	
Frequency of Use:	
Flow of Events:	1. User enters their email and password into the appropriate fields 2. User clicks on the `Sign In` button 3. System validates the account by matching the entered credentials with the database 4. System authenticates the user to sign-in successfully and is redirected to the Home page
Alternative Flows:	AF1: Sign with External Provider (Google) 1. Return to Step 4 AF2: Invalid credentials 1. System displays error message “Sign In Failed” 2. Return to Step 1 AF3: Invalid email input 1. System displays error message “Invalid Email”
Exception:	EX1: User not logged in to their External Provider (Google) account 1. System will prompt the user to input their External Provider (Google) account credentials to be authenticated
Includes:	1. Validate Credentials
Special Requirements:	

Assumptions:	
Notes and Issues:	

Use Case ID:	12		
Use Case Name:	Sign Up		
Created By:	Crystal Cheong	Last Updated By:	Ng Mu Rong
Date Created:	29/01/2023	Date Last Updated:	25/03/2023

Actor:	User, System
Description:	This use case describes the sign-up authentication process.
Preconditions:	- Device must be connected to the internet - User account must not already exist in the database
Postconditions:	- User is able to save listings - User is able to view account information - Account name is shown in the navbar
Priority:	
Frequency of Use:	
Flow of Events:	1. User enters their name and clicks `Next` 2. User enters their email and password into the appropriate fields 3. System validates the inputs before enabling the submission 4. User clicks on the `Sign Up` button 5. System validates the email address to ensure that its unique 6. System authenticates the user to sign-in successfully and is redirected to the Home page
Alternative Flows:	AF1: Sign with External Provider (Google) 1. Return to Step 5 AF2: Invalid credentials 1. System displays error message “Sign Up Failed” 2. Return to Step 1 AF3: Invalid email input 1. System displays error message “Invalid Email”
Exception:	EX1: User not logged in to their External Provider (Google) account 1. System will prompt the user to input their External Provider (Google) account credentials to be

	authenticated
Includes:	1. Validate Credentials
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	13		
Use Case Name:	Verify Credentials		
Created By:	Crystal Cheong	Last Updated By:	
Date Created:	29/01/2023	Date Last Updated:	

Actor:	User, System
Description:	This use case describes the session authentication process.
Preconditions:	- Device must be connected to the internet
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	1. System will check for the current authentication status and skip to step 5 if the active session is already authenticated 2. System will create a new User account in the database if one does not already exist and query the authenticated User account if it exists. 3. System will create a new Session using the account 4. System will also a create a new JSON Web Token (JWT) for the aforementioned Session 5. System will emit the current authenticated status and Session data
Alternative Flows:	AF1: User is already authenticated 1. Skip to step 5 AF2: User account is updated 1. System will query for the updated User account in the database 2. Proceed to step 3 AF3: User is signed out 1. System will destroy the authenticated Session and JWT

	token
Exception:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	14		
Use Case Name:	Via External Provider		
Created By:	Crystal Cheong	Last Updated By:	Ng Mu Rong
Date Created:	29/01/2023	Date Last Updated:	25/03/2023

Actor:	User, System
Description:	This use case describes the external provider authentication process.
Preconditions:	- Device must be connected to the internet - User has a supported external provider account - System supports at least one external provider
Postconditions:	- User is successfully authenticated
Priority:	
Frequency of Use:	
Flow of Events:	1. User selects `Continue with Google` button 2. User is already logged into their Google account 3. System will validate the account availability with respect to the database records 4. On the first time, the User will be prompted to grant permission for System to connect using User's Google account 5. Upon granting permission, User will be signed-in successfully using their Google account and is redirected to the Home page
Alternative Flows:	AF1: Invalid credentials 1. System displays error message "Sign Up Failed" 2. Return to Step 1
Exception:	EX1: User not logged in to their External Provider (Google)

	account 1. System will prompt the user to input their External Provider (Google) account credentials to be authenticated
Includes:	1. Validate Credentials
Special Requirements:	
Assumptions:	
Notes and Issues:	Supported External Providers: - Google

Use Case ID:	15		
Use Case Name:	Sign Out		
Created By:	Crystal Cheong	Last Updated By:	
Date Created:	29/01/2023	Date Last Updated:	

Actor:	User, System
Description:	This use case describes the sign-out process
Preconditions:	- Device must be connected to the internet - User account must already exist in the database - User is currently authenticated
Postconditions:	- User is signed out
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on `Sign Out` in the Account submenu 2. System will destroy the current Session and JWT token associated to the User 3. System will redirect to the Home page
Alternative Flows:	
Exception:	
Includes:	1. Verify Credentials
Special Requirements:	
Assumptions:	