Analysis Sequence Diagram Descriptions:

1. Log-in
   1. First, the user passes the Log-in information (email and password) to the UI Layer, which then asks the Data Layer to verify it. If it is verified, the Data Layer notifies the UI Layer, which displays a “Log-in Success” message to the User. If it is not verified, the Data Layer notifies the UI Layer, which displays a “Log-in Failed” message to the user.
2. Create Account
   1. First, the User taps the “New User” button. The UI Layer then tells the Data Layer to create a blank account. The Data Layer does so, and instructs the Domain Layer to begin its routine for collecting new user information. Next, the Domain Layer tells the UI Layer to display the set-up account page to the User, which it does. The User then enters their log-in information, as well as their roommate preferences, both of which are sent by the UI Layer to the Domain Layer. The Domain Layer bundles all this information and sends it to the Data Layer to be stored in the blank space created at the beginning of the process. If this is done successfully, the Data Layer confirms the account has been created to the Domain Layer, which informs the UI Layer to display an “Account Created!” message to the User. If this is not done successfully, the Data Layer informs the Domain Layer, which instructs the UI Layer to display an “Account Not Created Message” to the User. The UI Layer then does so.
3. Send a Chat
   1. First, the Sender enters a chat into the UI Layer. The UI Layer instructs the Domain Layer to access the established chat line between the Sender and the Recipient, which the Domain Layer does. The Domain Layer then uses this chat line to send the chat message to the UI Layer for the Recipient. Then, the UI Layer displays the chat on the Recipient’s screen and displays a message to the User saying a chat has been delivered.
4. Set Location
   1. First, the User selects the “Set Location” button on the UI Layer, which instructs the Data Layer to prepare Location Settings for Edits. After the Data Layer confirms it has done so, the UI Layer displays the location editing page to the User. The User enters their desired location, which the UI Layer transmits to the Data Layer. If the Data Layer successfully updates the location, it updates the UI Layer which then displays the “Location Updated” Message to the User. If the update is not successful, it informs the UI Layer which then displays the “Could not update” message to the User.
5. Personal Preferences Editing
   1. The User selects the “Edit Preferences” button and the UI Layer informs the Data Layer to open the User account for editing. The Data Layer confirms it has done so, and the UI Layer then displays the PReference Editing page to the User. Once the User enters the Preference Edits, the UI Layer passes them along to Data Layer which makes the desired edits. Afterwards, the Data Layer tells the UI Layer that the edits are complete and the UI Layer displays “Edits Saved” to the User.
6. Swipe Up on Other User
   1. First, the User swipes up on another user. Then, the UI Layer informs the Domain Layer that a swipe up has occurred. The Domain Layer then tells the Data Layer to add the Swipe Up to the User’s account, and then informs the UI Layer to display the swipe up animation to the User, which the UI Layer does. The Data Layer, after adding the Swipe Up to the User’s account,, informs the Domain Layer whether the User matched with the other user they swiped up on. If so, the Domain Layer informs the UI Layer, and the UI Layer displays a match notification to the User. If not, the Domain Layer informs the UI Layer, and the UI Layer displays a “No Match Yet” message to the User.
7. Swipe Down on Other User
   1. First, the Users swipes down on another User. Then, the UI Layer informs the Domain Layer that a Swipe Down has occurred. The Domain Layer then instructs the Data Layer to add the other user to the User’s account’s array of swiped down users. Afterwards, the Domain Layer tells the UI Layer to display the swipe down animation to the User, which the UI Layer does.

Design Sequence Diagram Descriptions:

1. Log-in
   1. First, the User passes their email and password to the UI Layer. The UI Layer then calls the Data Layer’s Verify function, which returns a verification boolean, with the email and password as parameters. If the boolean that Verify returns is a 1, the UI Layer then displays a success message to the User. If the boolean that Verify returns is a 0, the UI Layer then displays a failure message to the User.
2. Send a Chat
   1. First, the Sender passes a chat to the UI Layer. The UI Layer then calls the Domain Layer’s Access function, which returns a Chatline object, on the established Chat Line between the Sender and the Recipient. The returned Chatline object is stored in a variable called “Line”. The Domain Layer then calls the UI Layer’s Send function with both the chat and Line as parameters. Following this, the UI Layer calls its Deliver function, with the chat as parameter, to display the chat to the Recipient. The Deliver function returns a string, stored in the variable “msg”, that is a success message. The UI Layer then calls its Display function, with “msg” as the parameter, to display the success message to the Sender.
3. Set Location
   1. First, the User selects the “Set Location” button on the UI Layer. Next, the UI Layer uses the Data Layer’s Open function, which returns a string stored in the Status variable, with Location Settings as parameter. The UI Layer then sends the Status string back to the User, and the User sends their desired Location to the UI Layer. Then, the UI Layer calls the Data Layer’s Update function, which returns a String stored in the variable “msg”, with the Location Settings and the Location as parameters. The String stored in ‘msg’ is either a success or a failure message, depending on whether the update was a success. If the update was a success, the UI Layer displays “msg” (the success message) to the User; if the update failed, the UI Layer displays “msg” (the failure message) to the User.