RoomUp Design Class Justification and Description

**Log In:**

The classes that felt necessary to log in a user were the User, LogIn UI and the

Datalayer classes. The User and LoginUI classes are respectively connected to each other so that a user can enter information. Secondly, the User is connected to the DataLayer so that their logIn information can be verified in the database. Their information is either correct, and they are logged in or incorrect and they receive such a message displayed to them by the LoginUI.

**Send a Chat:**

The classes needed in order for a user to send a chat to another user were User, Sender and Recipient, ChatUI and Domain Layer. Sender and Recipient are both generalizations of users as they need UIDs and accounts to be able to chat with each other. For simplicity's sake in this DCD we have chosen not to include the aspect of needing to match with each other before having the ability to chat. Instead it is just assumed that the two users, sender and recipient, have matched with each other. Sender and Chat UI are both connected to each other so that the user can see and send messages. Likewise, the recipient is connected to Chat UI so that the recipient can see the message that the sender has sent. Lastly, there is a connection between Domain and ChatUI for the purpose of establishing and maintaining a chat line so two users can chat.

**Set up Account and Edit Preferences:**

The Datalayer, User, Account UI, Domain Layer and Account controller classes are

involved in setting up a user account and preferences. Here, in this DCD, The DataLayer has a connection to the User class because it needs account information to both set up an account and edit preferences. The AccountUI is Connected to the Domain Layer, Datalayer and AccountController so that it can provide all essential interfaces to the user so they can input and edit information needed to create or edit a profile. The Domain layer exists in this case to actually take the information from the user. Finally the account controller is responsible for actually creating the account and preferences attached to the account.

**Swipe and Match with a user:**

There are two “sets” of classes involved in swiping and matching with a user. The first set is everything that involves a user and the information about them. These classes includeUser and its subclasses hasRoom and needsRoom and additionally Profile and Preferences. The second section is everything needed to set up and determine a match between two users.

Match UI, matchController and DataLayer are the classes needed to do so. MatchUI is connected to and uses profile and preferences to display to a user, so they can choose if they want to match with them. Match UI also is connected to Datalayer so once a user has chosen to match or not match with another user, it can be recorded and saved. The match controller is used to determine, based on preferences, if two accounts should be suggested to each other as a match, as well as controlling what to do to each account once they have matched with each other.