













CRC overview

 Name	 Comments	 Methods	 ParentClasses	 Relationships	 Status 1
<u>Card 2</u>					
<u>Session</u>	Creates a login and runs. Creates SystemInOut			Swipe Options	UI
<u>Profile</u>	Different users informations. Bio, location etc. Sent Messages.				Entity
<u>Database</u>	Stores all the users. Create and store a list of potential matches (based on Given User's Preference) Pop a User from the potential matches and return relevant information (in string form).				Use Case
<u>Message</u>	Has content + sender and receiver.				Entity
<u>Thread</u>					Entity
<u>Current User</u>	Holds Database and does things with it.				Use Case

Aa Name	☰ Comments	☰ Methods	☰ ParentClasses	☰ Relationships	▼ Status 1
<u>MessageView</u>	<p>Constructor accepts a database and CurrentUser. Run() ⇒ Call ShowThreads().</p> <p>ShowThreads() ⇒ Asks CurrentUser to display all the Threads related to this user</p> <p>SelectThread(int) ⇒ Asks CurrentUser to display all messages in the thread.</p> <p>SendMessage(String) ⇒ AsksCurrentUser to create a new Message with the content in the String, with the sender and receiver based on the Thread.</p> <p>ShowMatches() ⇒ Ask CurrentUser to show all matches.</p> <p>SelectMatch(int) ⇒ Asks CurrentUser to create a thread between the current user and the selected match.</p> <p>ChangeView<View>() = accepts a type of view then creates an instance of that view and calls that views Run() method.</p>		View		Controller

Aa Name	Comments	Methods	ParentClasses	Relationships	Status 1
<u>SwipeView</u>	<p>Constructor accepts a database and CurrentUser. Run() ⇒ Asks DataBase to create a list of profiles and gets the top one to display. GetNextUser() ⇒ Asks DataBase for the next User. Swipe Right/Left() ⇒ Ask CurrentUser and DataBase to record a new like. Then GetNextUser(). ChangeView<View>() = accepts a type of view then creates an instance of that view and calls that views Run() method.</p>		View		Controller
<u>ProfileView</u>	<p>Constructor accepts a database and CurrentUser (from login). Run() ⇒ shows the information from CurrentUser and allows you to call the change/get methods. Send Change/get info requests to CurrentUser (which changes bio, age etc. on his profile) ChangeView<View>() = accepts a type of view then creates an instance of that view and calls that views Run() method.</p>		View		Controller
<u>View</u>		<p>ChangeView</p> <p>Constructor(Database User)</p> <p>Run</p>			Controller

 Name	 Comments	 Methods	 ParentClasses	 Relationships	 Status 1
<u>Login</u>	Calls run which asks you to either sign in or create account. In either case you eventually get to the point where you input a CurrentUser (who signed in) and a database of users into ProfileView. Then call the run for ProfileView.	Run			Controller
<u>CreateAccount</u>					Use Case