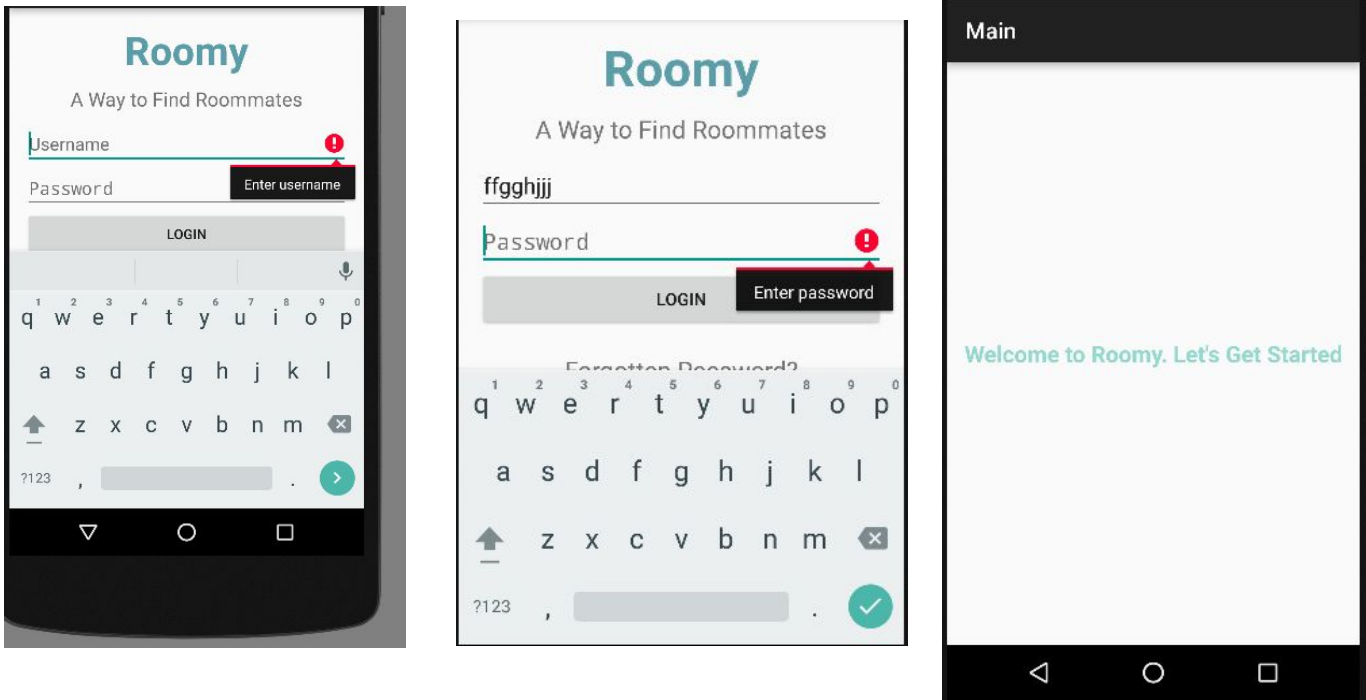


- Who: Jayce Meyer, Hunter Haskins, Evan Brookens, Elijah Staple
- Title of project: Roomy
- Vision: A simple meet & rating app for users to find information about nearby people looking for roommates
- Automated Tests:
 - Empty User Name Field
 - Empty Password Field
 - Login Successfully When User and Password are Correct
- Link: <http://mockito.org/>
- Explanation: We tested our app using Mockito with a Junit runner that is integrated into Android Studio and presents the data in an HTML. You can try and run it in android studio if you clone the git test repository, and then run the test class in main. The results were as follows:

LoginPresenterTest: 3 total, 3 passed		44 ms
		Collapse Expand
LoginPresenterTest		44 ms
ShowMessage_UsernameIsEmpty		passed 43 ms
StartMainActivity_UsernameAndPasswordWorked		passed 1 ms
Disconnected from the target VM, address: '127.0.0.1:44245', transport: 'socket'		
ShowMessage_PasswordIsEmpty		passed 0 ms



- User Acceptance Tests:

Use Case ID:	UC-01
Use Case Name:	Sign-out
Description:	A logged in user can click the sign-out button to log out of their account
Users:	App Member

Pre-conditions:	User Logged in		
Post-conditions:	User not logged in, and on home screen of app		
Frequency of Use:	Occasional by app members		
Flow of Events:	Actor Action	System Response	Comments
	1. Touch the Sign-out button		
	2. Confirm the wish to sign-out	Prompt the user with a yes and a no button to confirm their decision	
	3.	Direct the now logged out user to the home page of the app	After confirming the wish to sign-out, no further action is required from the user for the test to complete
Test Pass?:	Pass / Fail		
Notes and Issues:			

Use Case ID:	UC-02
Use Case Name:	Check for valid e-mail address
Description:	Passes test if user enters a valid email address (johndoe@example.domain)

Users:	Customers/Roomy Users		
Pre-conditions:	User must have valid email, and app needs to be able to receive the submit request		
Post-conditions:	Submission of email passes through successfully when the email is valid		
Frequency of Use:	Every time a new user creates an account		
Flow of Events:	Actor Action	System Response	Comments
	1. Click on Sign Up	Opens box to add email and password	
	2. Enter email/sign up info		
	3. Submit entries	Algorithm checks validity of email before submit goes through successfully	Password matching would be a different test program/algorithm
Test Pass?:	Pass / Fail		
Notes and Issues:			

--	--

Use Case ID:	UC-03
Use Case Name:	Create account
Description:	User can successfully enter valid email and password to create an account

Users:	Roomy app users/customers		
Pre-conditions:	User has valid email and matching passwords to create and gain access to an account		
Post-conditions:	User can enter their newly created account with the email and password they used to create it		
Frequency of Use:	Used every time a new user wants to create an account on Roomy, and also goes hand in hand with successfully logging in to user accounts		
Flow of Events:	Actor Action	System Response	Comments
	1. User clicks Sign Up	Opens new window for email/password input	
	2. User passes in email and password	System runs Valid email algorithm, checks that passwords match, if both are successful the account is created with the given credentials	
	3. User is given a Success/Verification page	System logs account info and (assuming we use a hash table) adds the account to the database	
Test Pass?:	Pass / Fail		
Notes and Issues:			

Use Case ID:	UC-04
Use Case Name:	Rate Roommate
Description:	While viewing a roommates profile, the user can click on the rate button and provide information on that user and submit the info

Users:	App members
Pre-conditions:	User is logged in and is viewing the profile page of one of their current or past roommates
Post-conditions:	The user's review is submitted and is shown on the reviews section of the roommate they reviewed
Frequency of Use:	Occasionally by members of Roomy

Flow of Events:	Actor Action	System Response	Comments
	1. Click on the review button on the profile page of the roommate they wish to review.	Bring user to a review page, in which all of the categories are displayed for the user to provide a rating on.	
	2. Provide the information they wish to provide in categories given	Update the review page to reflect the information that the user has provided	This must be done to show the user what they have already entered and what they haven't
	3. Click on the submit button to finalize the review.	Store the information the user provide and have it shown on the reviews section of the person reviewed.	
Test Pass?:	Pass / Fail		
Notes and Issues:			

Use Case ID:	UC-05
Use Case Name:	Password Match
Description:	Algorithm that makes sure passwords match when creating an account

Users:	New users who are creating an account		
Pre-conditions:	User has a valid email and does not already have an account on Roomy		
Post-conditions:	User passes valid email check, then passes password match and successfully creates an account		
Frequency of Use:	Every time a new user creates an account		
Flow of Events:	Actor Action	System Response	Comments
	1. Click Sign Up from Login page	System opens new window for user to input email and password	Password will be entered twice so the user doesn't accidentally set a password they didn't mean to type
	2. User submits email and password	System first checks validity of email, then run match algorithm on both password entries	This algorithm would be a simple loop that checks each character (we will have to make sure this is a safe option since we're working with passwords)

	3.Successfully matches passwords	Algorithms finishes successfully and goes to a page that notifies the user that their account is live	
Test Pass?:	Pass / Fail		
Notes and Issues:			

- VCS : <https://github.com/evbr1432/Roomy>