

Test Plan - "1/4 Cup?"

CS408 Software Testing - Spring 2018

Jimmy Carlson, Janka Gál, Puja Mittal, Jason Shipp, David Whorley

#1: Login-Proceed Screen

1. Functional

- A. Login-Proceed 001, Button Click, Severity 1
 - B. Click "Sign In" button to advance to the next screen.
 - C. Click is registered and redirected to a sign in page with Google and email registration.
-
- A. Login-Proceed 002, Random Screen Click, Severity 2
 - B. On Login-Proceed screen use Android input fuzzer Exerciser Monkey to register random clicks on the screen at an increased rate, avoiding sign-in.
 - C. Nothing should occur.
-
- A. Login-Proceed 003, Window to Android Home Page, Severity 2
 - B. Click the circle Android home button and return to homepage, then click the Android square window button and reopen the "1/4 Cup?"
 - C. The user should land back at the same page the exited from.

#2: Login-Registration Screen

1. Functional

- A. Login-Proceed 001, "Sign in with email" Button Press, Severity 1
 - B. Click the "Sign in with email" button.
 - C. Proceed without error to the email login screen.
-
- A. Login-Proceed 002, "Sign in with Google" Button Press, Severity 1
 - B. Click the "Sign in with Google" button.
 - C. Proceed without error the the Google/GMail Account login screen.

- A. Login-Proceed 003, Back Button Click, Severity 2
 - B. Click the android interface back button
 - C. The user should land back at the Login-Screen with no error
-
- A. Login-Proceed 004, Random Screen Click, Severity 2
 - B. On Login-Registration screen use Android input fuzzer Exerciser Monkey to register random clicks on the screen at an increased rate, avoiding "Sign in with email" and "Sign in with Google" buttons.
 - C. Nothing should occur.
-
- A. Login-Proceed 005, Window to Android Home Page, Severity 2
 - B. Click the circle Android home button and return to homepage, then click the Android square window button and reopen the "1/4 Cup?"
 - C. The user should land back at the same page the exited from.

#3: Email-Sign-In Screen

1.Functional:

- A. Email-Sign-In 001, Regular Sign In, Severity 1
 - B. User inputs their own email, First/Last Name, and Password and clicks sign-in
 - C. User should be passed towards the functional portion of the app without issue.
-
- A. Email-Sign-In 002, Incorrect Sign In, Severity 2
 - B. User inputs incorrect, but still correctly formatted, Email account information for login, after having logged in a previous time.
 - C. User is informed that the information is incorrect.
-
- A. Email-Sign-In 003, Back Button Click, Severity 2
 - B. Click the android interface back button.
 - C. The user should land back at the Email-Sign-In Screen with no error
-
- A. Email-Sign-In 004, Random Screen Click, Severity 2
 - B. On Email-Sign-In screen use Android input fuzzer Exerciser Monkey to register random clicks on the screen at an increased rate, avoiding the sign in button.

- C. Nothing should occur.

- A. Email-Sign-In 005, Window to Android Home Page, Severity 2
- B. Click the circle Android home button and return to homepage, then click the Android square window button and reopen the "1/4 Cup?" screen.
- C. The user should land back at the same page the exited from.

2.Equivalence Class:

- A. Email-Sign-In 006, Non Valid Text Email, Severity 2
- B. User inputs a non-numerical input in to the login box with no @ sign included to denote a non-valid email.
- C. User is prompted that the email sign in did not work or is invalid.

- A. Email-Sign-In 007, Non Valid Numerical Email, Severity 2
- B. User inputs only numerical values to the login box.
- C. User should be prompted that sign in did not work.

- A. Email-Sign-In 008, Emoticons in Fields, Severity 2
- B. User uses Android "Emojis" in the three fields during their login attempt.
- C. User is prompted that the login is invalid.

- A. Email-Sign-In 009, One Name First/Last Name, Severity 2
- B. User only inputs a single text field in the First/Last Name box.
- C. User is told the field is invalid.

Boundary Value:

- A. Email-Sign-In 009, Blank Password Field, Severity 2
- B. User fills out a valid email but does not input a password then attempts submission.
- C. User is notified that the password field is empty.

- A. Email-Sign-In 010, Blank Username Field, Severity 2
- B. User fills out the password field but not the username field and attempts submission.
- C. User is notified that the username field is empty.

- A. Email-Sign-In 011, Both Fields Blank, Severity 2
 - B. User just hits the submit button.
 - C. User should be notified that the fields are blank.
-
- A. Email-Sign-In 012, Overflow, Severity 3
 - B. User dramatically overfills the username, password, and first/last name field (100+ characters).
 - C. User should be informed the input is invalid as normal.

#4: Google-Sign-In Screen

1.Functional:

- A. Google-Sign-In 001, Regular Sign In, Severity 1
 - B. User inputs their Google account information for login.
 - C. User is logged in successfully and advances to the actual functional portions of the app.
-
- A. Google-Sign-In 002, Incorrect Sign In, Severity 2
 - B. User inputs incorrect, but still correctly formatted, Google account information for login.
 - C. User is informed that the information is incorrect.
-
- A. Google-Sign-In 003, Back Button Click, Severity 2
 - B. Click the android interface back button.
 - C. The user should land back at the Login-Registration Screen with no error
-
- A. Google-Sign-In 004, Random Screen Click, Severity 2
 - B. On Login-Registration screen use Android input fuzzer Exerciser Monkey to register random clicks on the screen at an increased rate, avoiding the sign in button.
 - C. Nothing should occur.
-
- A. Google-Sign-In 005, Window to Android Home Page, Severity 2
 - B. Click the circle Android home button and return to homepage, then click the Android square window button and reopen the "1/4 Cup?" screen.
 - C. The user should land back at the same page the exited from.

2. Equivalence Class:

- A. Google-Sign-In 006, Non Valid Text Email, Severity 2
 - B. User inputs a non-numerical input in to the login box with no @ sign included to denote a non-valid email.
 - C. User is prompted that the email sign in did not work or is invalid.
-
- A. Google-Sign-In 007, Non Valid Numerical Email, Severity 2
 - B. User inputs only numerical values to the login box.
 - C. User should be prompted that sign in did not work.
-
- A. Google-Sign-In 008, Emoticons in Fields, Severity 2
 - B. User uses Android "Emojis" in the two fields during their login attempt.
 - C. User is prompted that the login is invalid.

3. Boundary Value Tests:

- A. Google-Sign-In 009, Blank Password Field, Severity 2
 - B. User fills out a valid Google Account email but does not input a password then attempts submission.
 - C. User is notified that the password field is empty.
-
- A. Google-Sign-In 010, Blank Username Field, Severity 2
 - B. User fills out the password field but not the username field and attempts submission.
 - C. User is notified that the username field is empty.
-
- A. Google-Sign-In 011, Both Fields Blank, Severity 2
 - B. User just hits the submit button.
 - C. User should be notified that the fields are blank.
-
- A. Google-Sign-In 012, Overflow, Severity 3
 - B. User dramatically overfills the username and password field (100+ characters).
 - C. User should be informed the input is invalid as normal.

#5: My-Lists Screen

1. Functional

- A. My-Lists 001, Logout, Severity 1
- B. User clicks logout the right hand corner button then the logout button
- C. User is directed to the Login-Registration Screen

- A. My-Lists 002, Settings, Severity 1
- B. User clicks the right hand corner button then the settings button
- C. User is directed to the Settings Screen

- A. My-Lists 003, Swipe Right, Severity 1
- B. User swipes right on the My-Lists screen.
- C. User should be shown a panel with the recipe/list section (properly populated).

- A. My-Lists 004, Upper-Left Button, Severity 1
- B. User should tap the button on the upper left hand corner.
- C. User should be shown a panel with the recipe/list section (properly populated).

- A. My-Lists 005, Create a New List, Severity 1
- B. User clicks create new list option, and enters a valid text field into the box, before clicking "create new list"
- C. A blank list should populate the recipe/list section, and you should be directed to a screen titled with the text field.

- A. My-Lists 006, Delete List, Severity 1
- B. User swipes right to left on a list in the recipe/list panel.
- C. The list should be deleted from the panel.

- A. My-Lists 007, Click Recipes, Severity 1
- B. User clicks the recipe button in the recipe/list panel.
- C. The User should be directed to the recipe button.

2. Equivalence Class

- A. My-Lists 008, List Emoticons, Severity 2
- B. User inputs Android "emojis" in the list name pop up.

- C. The name should be accepted without issue and added to the panel.

3. Boundary Cases

- A. My-Lists 009, Overflow, Severity 2
- B. User significantly overfills list name pop up.
- C. The name should be accepted without issue and added to the panel.

#6: List Screen

1. Functional

- A. List 001, Add Valid Item, Severity 1
- B. User inputs a valid food item one would expect to find in a recipe.
- C. Item is added to the list without issue.

- A. List 002, Add Invalid Item, Severity 1
- B. User inputs a non-sensical real object that is not food based.
- C. Toast error is thrown to the user

- A. List 003, Check Item, Severity 1
- B. User swipes right on the item added to the list.
- C. Item is checked green to indicate it has been selected.

- A. List 004, Uncheck Item, Severity 1
- B. User swipes right on a checked item on the list.
- C. Item is returned to its normal color and is no longer selected.

List 005, Delete Item - Unchecked, Severity 1
User swipes left on a unchecked item on the list.
The item is removed from the list

- A. List 006, Delete Item - Checked, Severity 1
- B. User swipes left on a checked item on the list.
- C. The item is removed from the list and is not apparent on the checked item screens later.

2. Equivalency Class

- A. List 007, Emoji, Severity 2

- B. User inputs Android "Emoticons" into the item field.
- C. App gracefully throws the normal toast error for non valid food.

3. Boundary Case

- A. List 008, Empty Field, Severity 2
 - B. User inputs a blank field into the item search and submits it.
 - C. Toast error is thrown about an empty field without making a call
-
- A. List 009, Overfill, Severity 2
 - B. User dramatically overfills the item search and submits it.
 - C. Toast error is thrown about non valid food gracefully.
-
- A. List 010, API call limit, Severity 2
 - B. Automated calls for valid foods are made at more than 5 times a second, and over a 1000 times in a month.
 - C. User is gracefully thrown a toast error instead of crashes.

#7: Search Recipes

1. Functional

- A. Search 001, Empty Field, Severity 2
 - B. User inputs a blank field into the recipe search and submits it.
 - C. User should see a toast stating 'Please enter a search query'
-
- A. Search 002, Valid Field, Severity 1
 - B. User inputs "chicken" into the search bar
 - C. Results should show recipes that include the word "chicken"
-
- A. Search 003, Valid Field, Severity 1
 - B. User inputs "french toast" into the search bar
 - C. Results should show recipes that include the words "french toast"
-
- A. Search 004, Select Recipe, Severity 1
 - B. User selects a recipe from the list
 - C. "Add A Recipe" window is displayed

2. Equivalency Class

- A. Search 005, Invalid Input, Severity 2
- B. User types in an emoji into the search field
- C. User should see a toast stating 'HTTP Error'

- A. Search 006, Invalid Input, Severity 2
- B. User types "Bdfjdkj" into the search field
- C. User should see a toast stating 'HTTP Error'

3. Boundary Case

- A. Search 007, 5 API Calls, Severity 3
 - B. User types in 6 searches quickly within a minute to exceed the 5 API calls allowed per minute
 - C. User should see a toast stating 'API Error'
-
- A. Search 008, Overfill, Severity 2
 - B. User dramatically overfills the item search and submits it.
 - C. Toast error is thrown stating 'HTTP Error'

#8: Add Recipe

1. Functional

- A. Recipe 001, Select Shopping List, Severity 2
 - B. Select a shopping list from drop down menu and click "Add Recipe"
 - C. Window should switch to the selected shopping list and items should be added to it
-
- A. Recipe 002, Uncheck Items, Severity 1
 - B. Uncheck some items from recipe ingredients list and click "Add Recipe"
 - C. Window should switch to the selected shopping list and only checked items should be added to it

2. Equivalency Class

- A. Recipe 003, Uncheck All Items, Severity 2
- B. Uncheck all items from recipe ingredients list and click "Add Recipe"

- C. The user is prompted with a toast stating “you must select at least one ingredient.”

#9: Sort Orders

1. Functional

- A. Sort Order 001, Sort, Severity 2
- B. User clicks on a category to sort by
- C. List should show up on the Test List screen sorted by selected category

3. Boundary Case

- A. Sort Order 002, No Categories, Severity 2
- B. User clicks on “Sort Order” on upper-right button when there are no categories set
- C. Toast error is thrown gracefully about needing to add categories to sort with

#10: Categories

1. Functional

- A. Categories 001, Add Category, Severity 2
 - B. User clicks the plus button
 - C. User should see an “Add Category” toast pop up
-
- A. Categories 002, Sort Categories, Severity 1
 - B. User drags and drops to reorder the categories
 - C. User should see the reordered list on the categories screen
-
- A. Categories 003, Add Category - Valid Category, Severity 2
 - B. User inputs a valid category
 - C. User should see the category on the Categories screen
-
- A. Categories 004, Delete Category, Severity 1
 - B. User swipes on a category to delete the category
 - C. User should no longer see that category on the categories list
-
- A. Categories 005, Random Screen Click, Severity 2

- B. On Test Sort Order screen use Android input fuzzer Exerciser Monkey to register random clicks on the screen at an increased rate, avoiding "+" and any categories
 - C. Nothing should occur.
-
- A. Categories 007, Assigning Categories, Severity 3
 - B. On Test List screen, user should be able to hold down on an item in the list to assign them to a user-selected category
 - C. Selected item should have the selected category
-
- A. Categories 008, Add multiple items, Severity 3
 - B. User creates a category, and adds multiple items
 - C. After entering multiple categories, the user should be able to re-arrange them to change the 'priority' of certain categories

2. Equivalency Class

- A. Categories 008, Add Category, Severity
- B. User inputs an invalid category name
- C. App gracefully throws the normal toast error for non valid category

3. Boundary Case

- A. Categories 009, Empty Field, Severity 2
 - B. User inputs a blank field into the "Add Category" field and submits it.
 - C. Toast error is thrown about an empty field without making a call
-
- A. Categories 010, Overfill, Severity 2
 - B. User dramatically overfills the "Add Category" field and submits it.
 - C. Toast error is thrown about non valid category name gracefully.