Test Plan Document

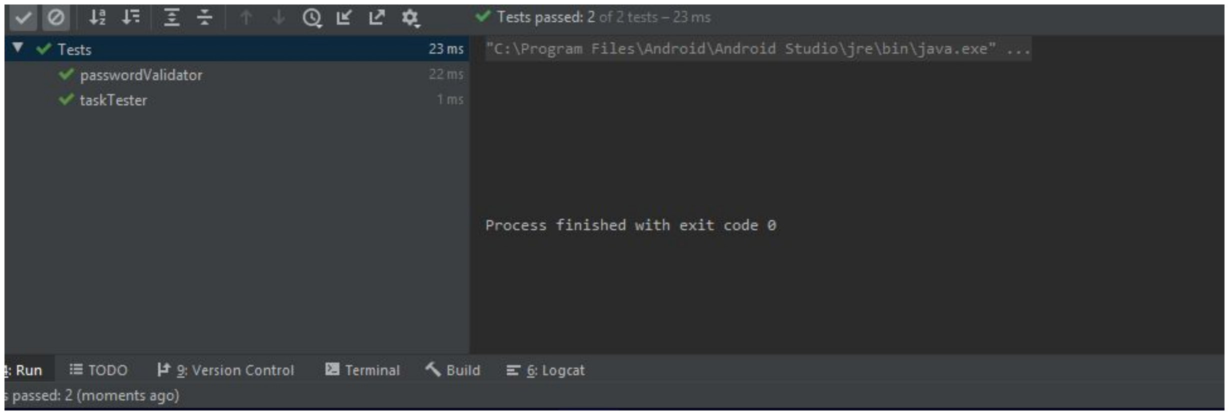
For *Teaching Tasks* App

Unit Tests

```
import android.widget.Button;

import com.example.teachingtasks.CreateUserEventHandler;
import com.example.teachingtasks.Task;

import org.junit.*;

import java.util.HashMap;
import java.util.UUID;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertFalse;
import static org.junit.Assert.assertTrue;

public class Tests {
   @Test
   public void passwordValidator() {
       CreateUserEventHandler c = new CreateUserEventHandler();

       //Valid
       String good = "Abc123@@";

       //Invalid
       String len_7 = "Abc123@";
       String no_cap = "abc123@@";
       String no_letter = "123456@@";
       String no_special = "Abc12345";
       String no_number = "Abcdef@@";

       assertTrue(c.isAcceptablePassword(good));

       assertFalse(c.isAcceptablePassword(len_7));
       assertFalse(c.isAcceptablePassword(no_cap));
       assertFalse(c.isAcceptablePassword(no_letter));
       assertFalse(c.isAcceptablePassword(no_special));
       assertFalse(c.isAcceptablePassword(no_number));
   }

   @Test
   public void taskTester() {
       UUID id = new UUID(100L, 50L);
       HashMap<String, Button> hm = new HashMap<>();
```

```
        Task t = new Task(id, "question",hm);

        t.setQuestionObject("new_question");
        t.setMastery(2);

        assertEquals(id, t.getTaskID());
        assertEquals("new_question",t.getQuestionObject());
        assertEquals(2, t.getMastery());

    }

}
```

RESULTS:


Tests passed: 2 of 2 tests – 23 ms

```
Tests                23 ms
  passwordValidator  22 ms
  taskTester          1 ms
```

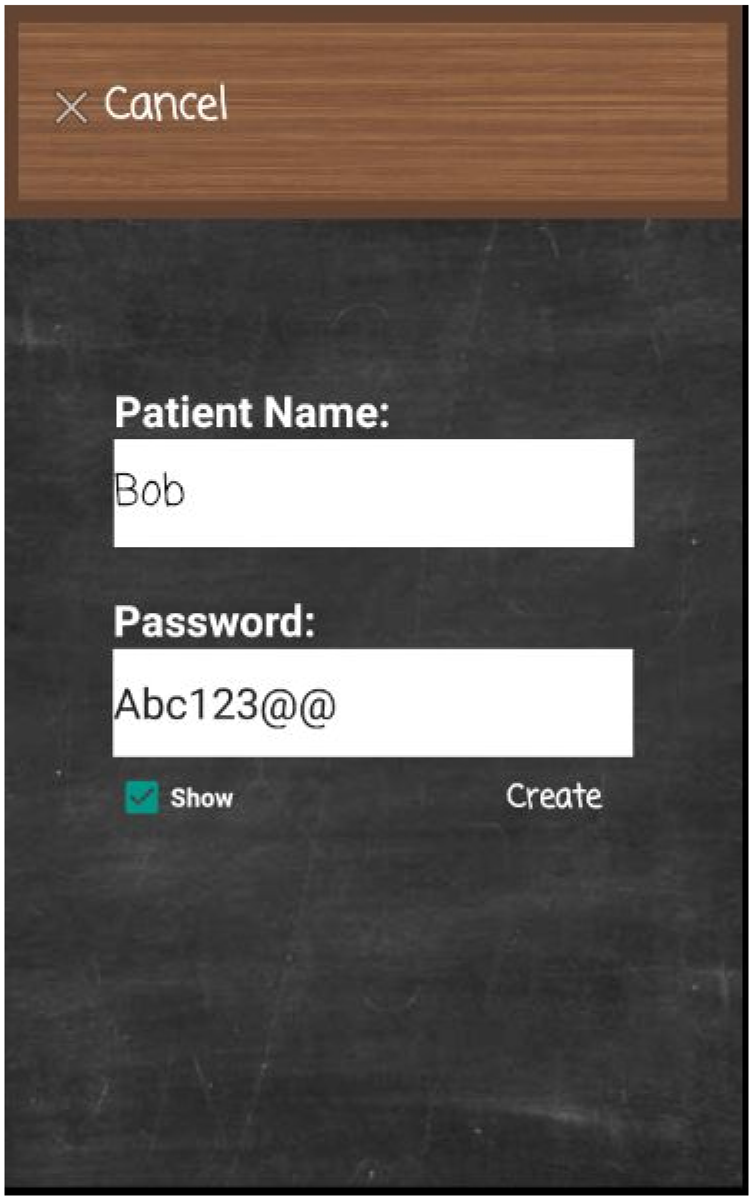
```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...


Process finished with exit code 0
```

passed: 2 (moments ago)

Use-Case Tests

Title: Create User
Actors: User
Requirements: User is on the login GUI
Main Scenario: 1) User opens app 2) User selects “create user” button 3) System goes to the user creation page 4) User types in name 5) User types in password, that must contain letters, numbers, and special characters 6) User selects “done” button

7) System creates a user 8) System adds user to the user selection page 9) System returns to the user selection page
Alternatives: 4a) User types in a name that already exists 4a1) System rejects name 4a2) System says user already exists 5a) User types a password not containing letters, numbers, and special characters 5a1) System rejects password 5a2) System says that passwords must contain letters, numbers, and special characters.
Test Situations: 1) User enters a valid password 2) User enters an invalid password 3) User enters a new name 4) User enters a preexisting name
Results: **PASS**



Patient Name:

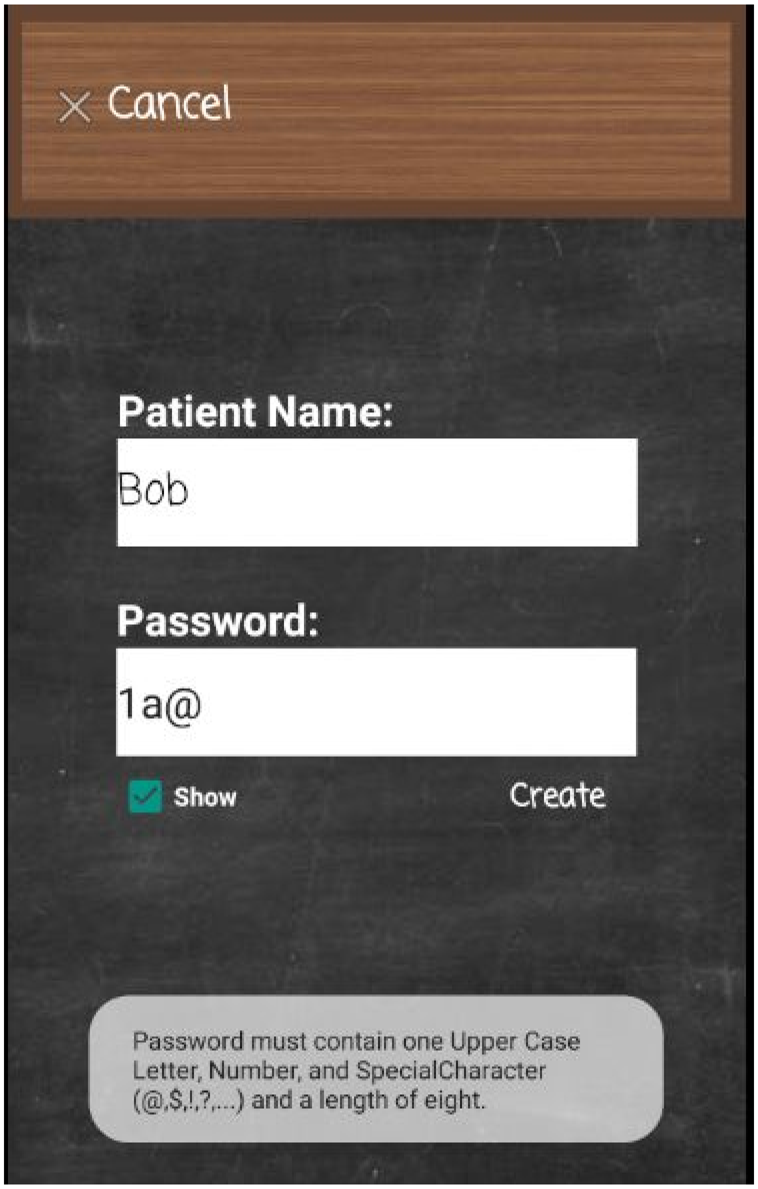
Bob

Password:

Abc123@@

✓ Show

* Results of test situations 1 & 2



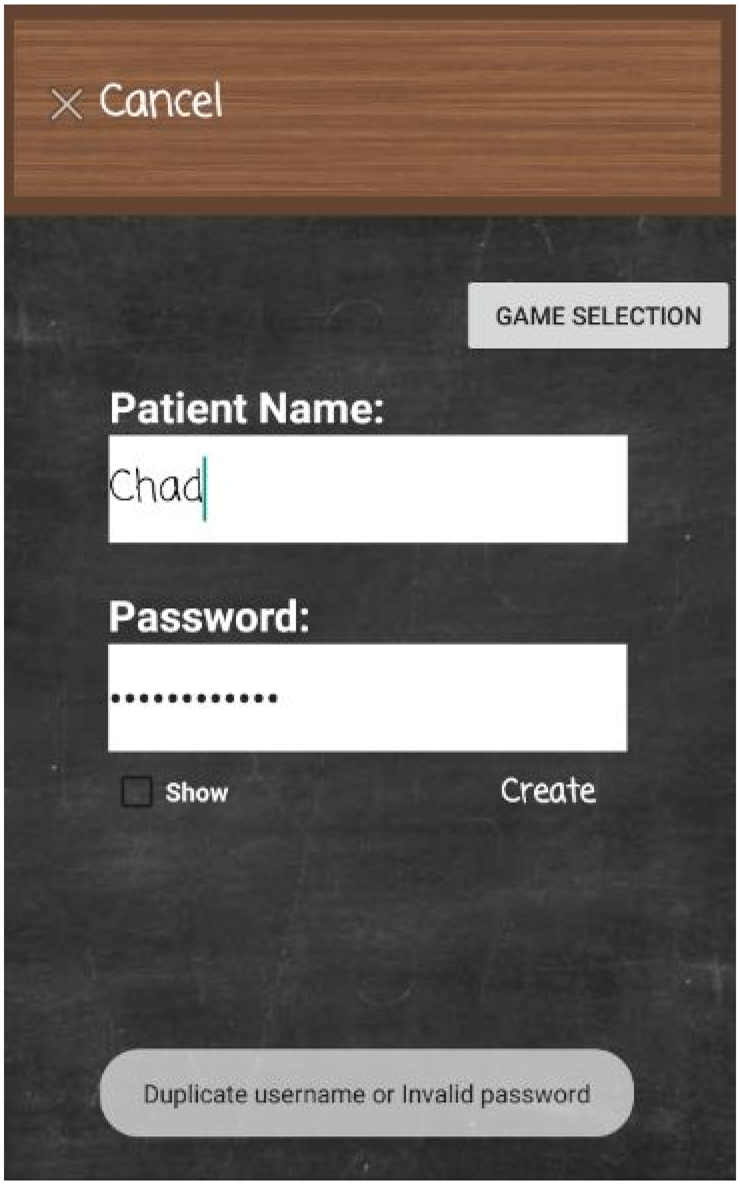
Bob

1a@

✓ Show

Password must contain one Upper Case Letter, Number, and SpecialCharacter (@,$,!,?,...) and a length of eight.

* Results of test situation 2



Duplicate username or Invalid password

* Results of test situation 4

Title: Delete User
Actors: User
Requirements: User is on the login GUI
Main Scenario: 1) User selects the “Edit” button

2) User selects the accounts to delete
3) User hits the “Delete” button
4) System deletes Users from the database

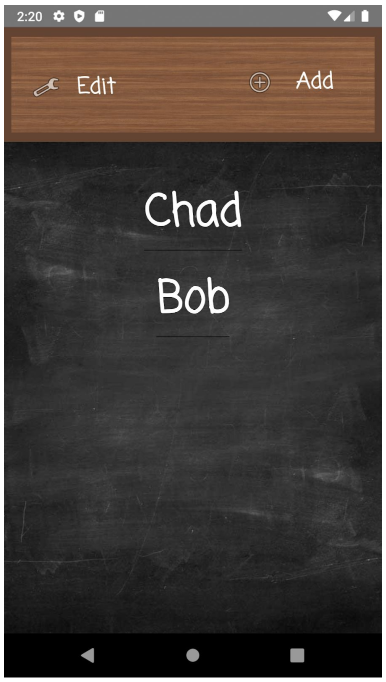
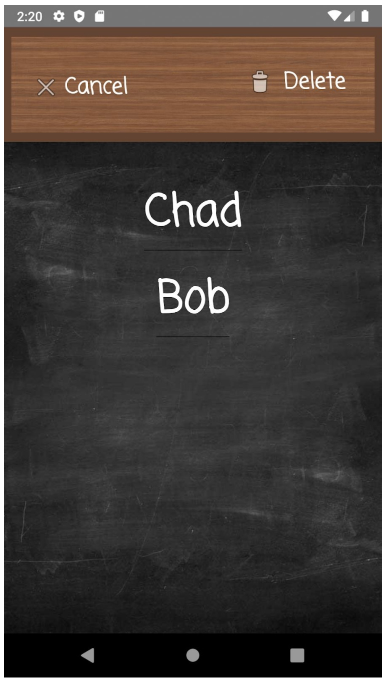
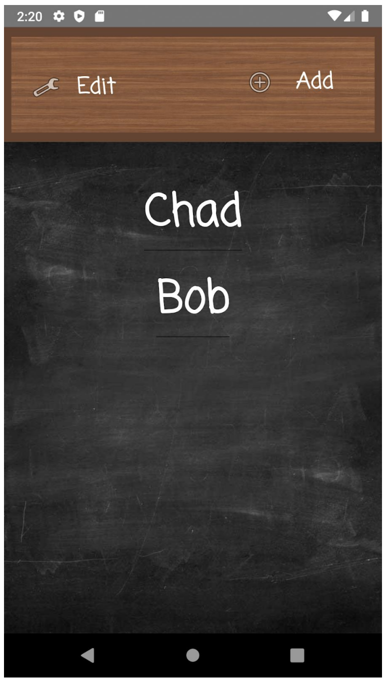
Alternatives:

3a) User hits “Cancel” button
3a1) System does not delete the selected accounts
3a2) System returns to the login GUI

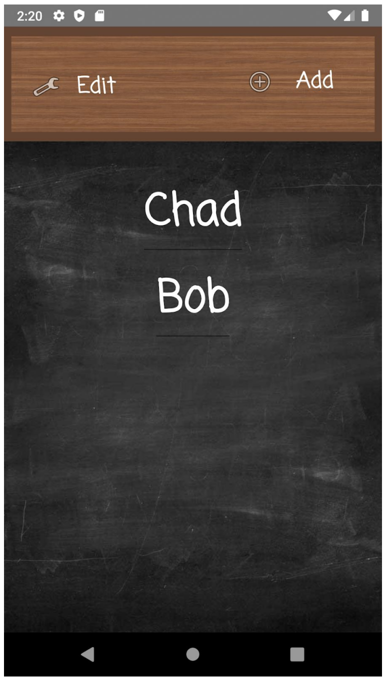
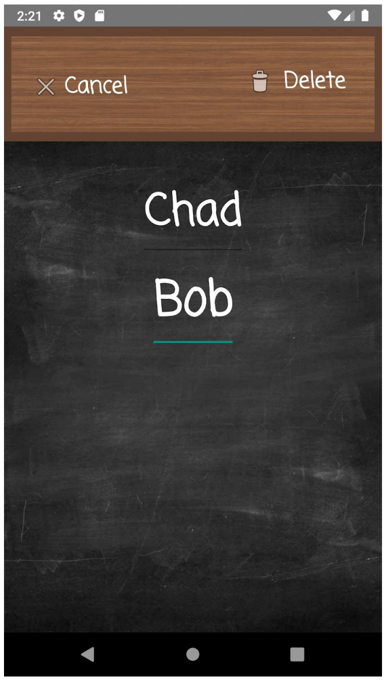
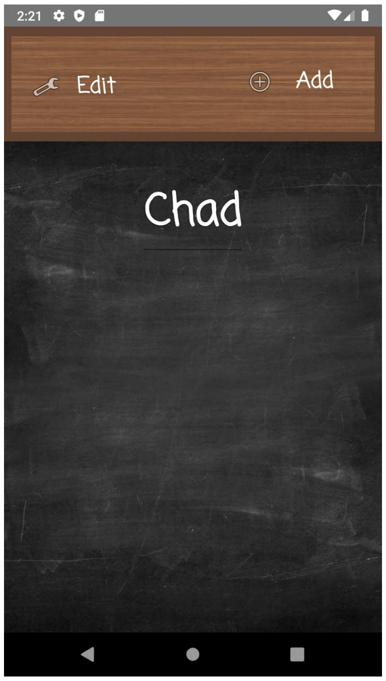
Test Situations:

1) User decides to not delete a user
2) User deletes a user

Results: **PASS**

* Results of test situation 1 (User hit “edit”, then “cancel”)

* Results of test situation 2

Title: User login

Actors: User

Requirements: User account exists

Main Scenario:

1) User types in password
2) User hits “login” button
3) System logs in as user

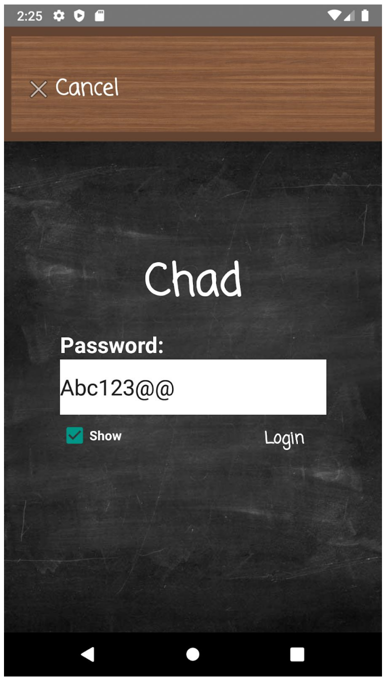
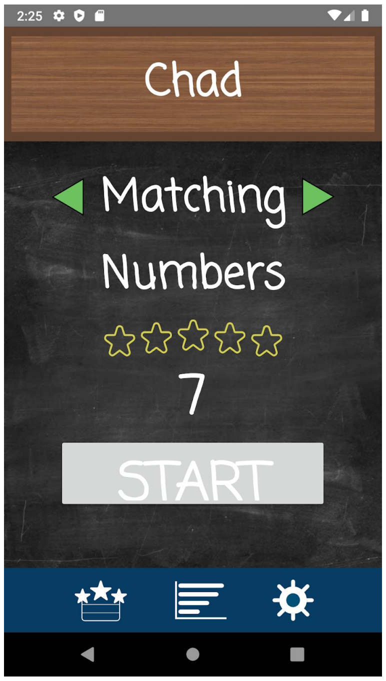
Alternatives:

1a) User types an incorrect password
1a1) System does not log user in
1a2) System says password was incorrect

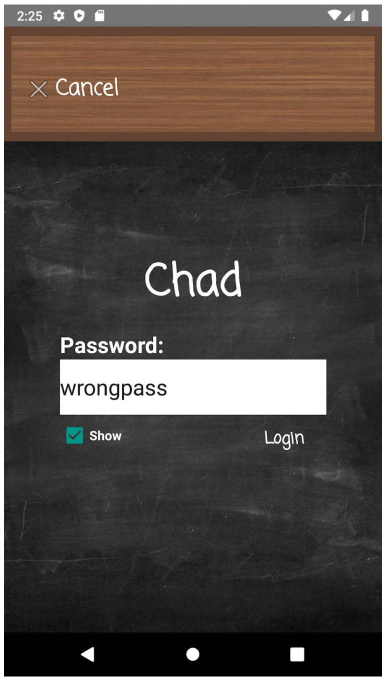
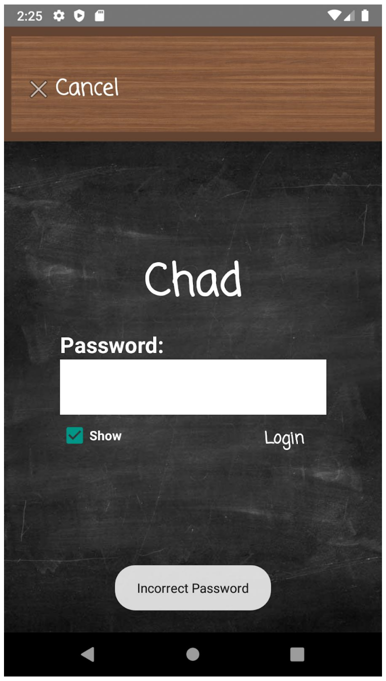
Test Situations:

1) User types the correct password
2) User types the incorrect password

Results: **PASS**

 Password: Abc123@@		

* Results of test situation 1

 Password: wrongpass	 Incorrect Password	

* Results of test situation 2

Title: User logout

Actors: User

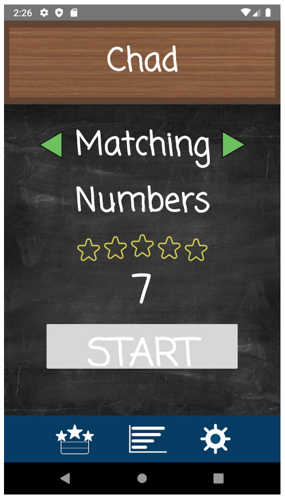
Requirements: User is logged in, user is in the settings GUI

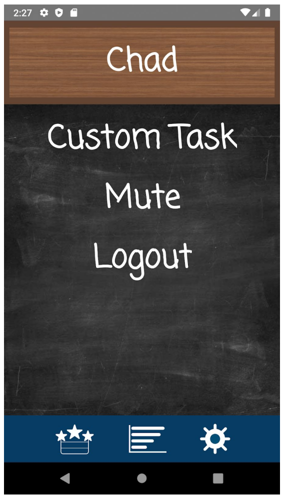
Main Scenario:
1) User selects logout
2) System logs out as user
3) System returns user to login GUI

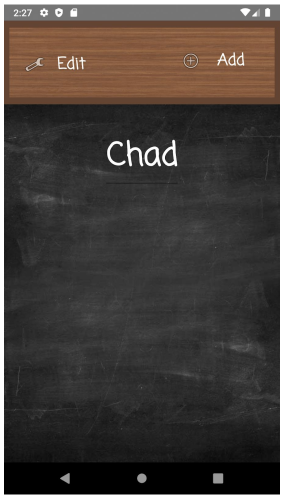
Alternatives:
3a) User hits the phone's "back" button
3a1) System does not permit user to go back

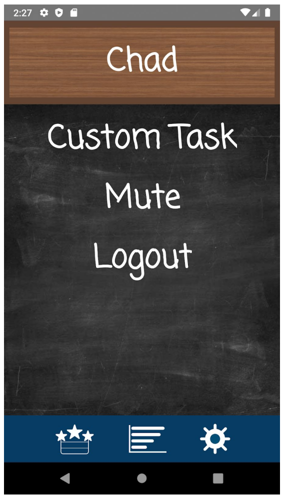
Test Situations:
1) User tries to go back

Results: **FAIL**









* Results of test situation 1 (user presses system's back button on photo 3)

Title: Gameplay

Actors: User

Requirements: User is logged in

Main Scenario:

1) System adds objects to screen
2) System says to tap the correct object
3) User taps an object
4) System displays results

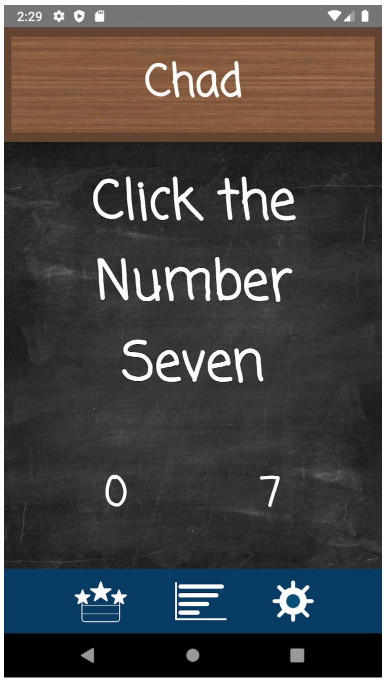
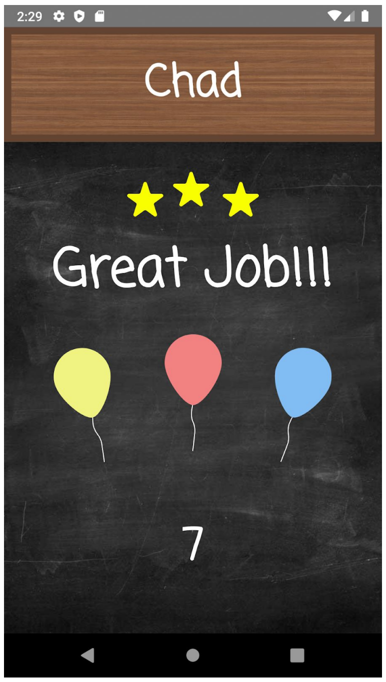
Alternatives:

3a) User taps incorrect object
3a1) System does not display results screen
3a2) System says that the object tapped is incorrect

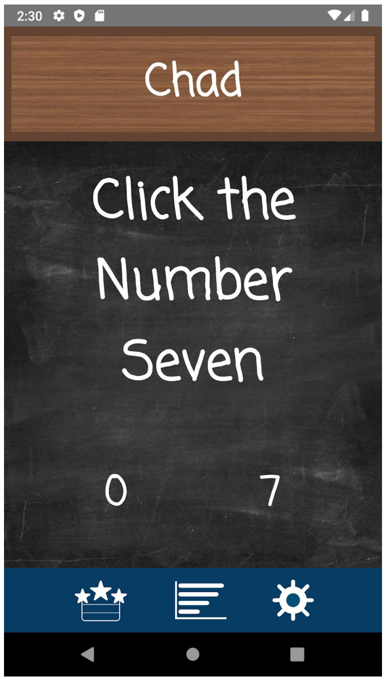
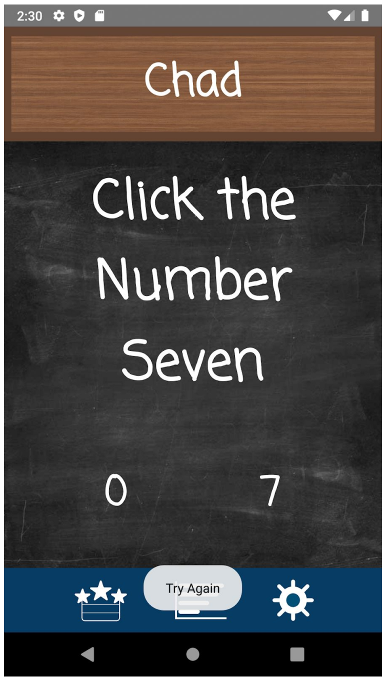
Test Situations:

1) User taps the correct object
2) User taps an incorrect object

Results: **PASS**

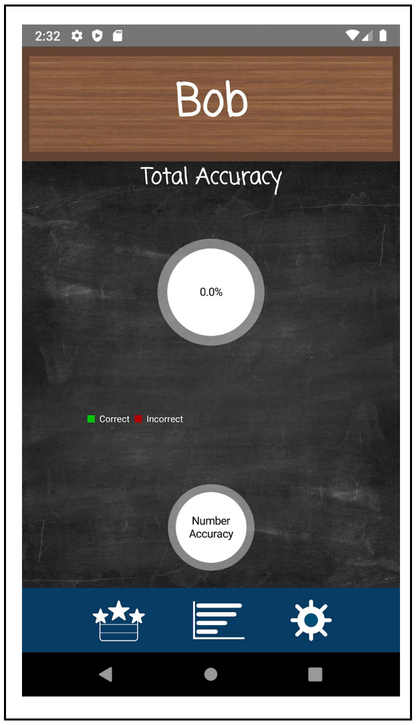
 2:29 Chad Click the Number Seven 0 7	 2:29 Chad Great Job!!! 7	

* Results of test situation 1

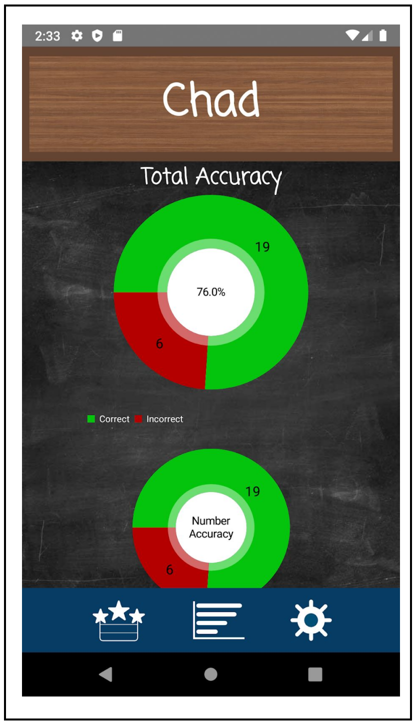
	 2:30 Chad Click the Number Seven 0 7 Try Again	

* Results of test situation 2

Title: View Statistics
Actors: User
Requirements: User is logged in
Main Scenario: 1) User selects the “View Statistics” button 2) System shows pie charts of correct and incorrect answers
Alternatives: 2a) Pie charts have 0 correct and 0 incorrect answers 2a1) The pie chart displays as “0.0% incorrect”
Test Situations: 1) Pie chart data has 0 correct and 0 incorrect answers 2) Pie chart data has a non-zero amount of correct and incorrect answers
Results:



* Results of test situation 1



* Results of test situation 2

Acceptance Tests

Acceptance Criteria:

- Encrypted passwords
- Secure log-in
- Secure log-out
- Functional game
- Visual statistics for game progress

Acceptance Test Plan:

Encrypted Password Testing:

- Validate that the source code uses a hashing algorithm to encrypt passwords
- Crack the application's data and verify that passwords are not stored as plaintext and as encrypted text.

Secure log-in Testing:

- Validate that the user cannot login with an incorrect password
- Validate that the user can login with a correct password

Secure log-out Testing:

- Validate that the user cannot press the system's back button to access the user's account.

Functional Game Testing:

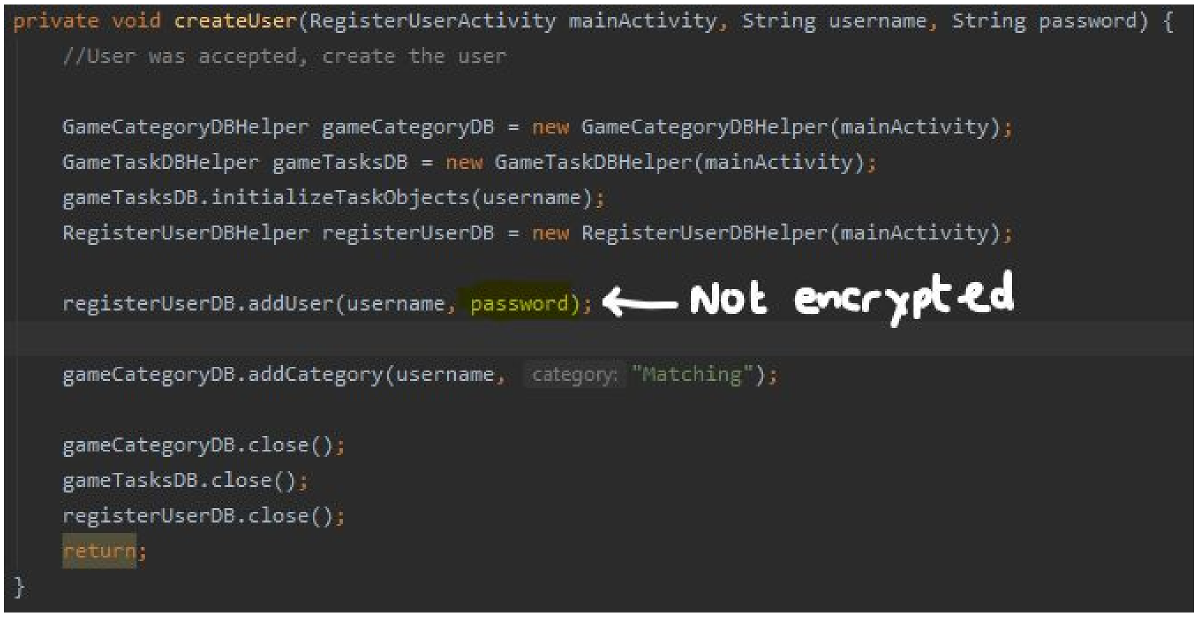
- Play 50 rounds of choosing a mixture of correct and incorrect answers
    - Validate that the game only progresses on correct answers

Visual Statistics Testing:

- Play 20 rounds of the game choosing a mixture of correct and incorrect answers
    - Do a paper calculation to see if the statistics percentages are match
- Play 10 rounds of the game choosing only correct answers
    - Validate the the statistics say 100% correct
- Create a new user
    - Validate that the statistics say 0% incorrect and 0% correct.

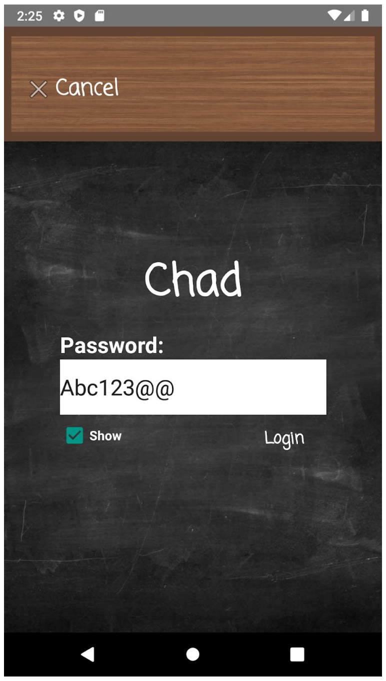
Test Results: **FAILED**

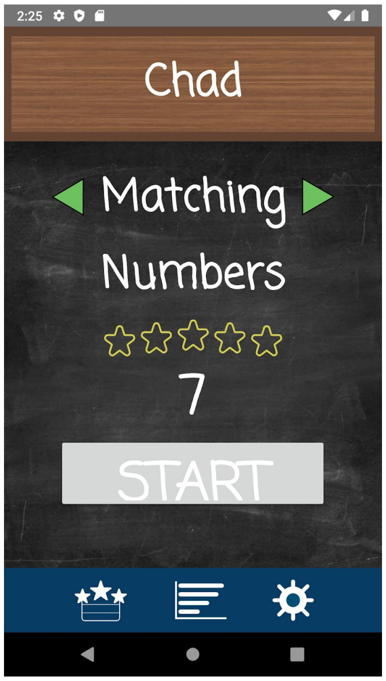
Encrypted Password Testing: **FAILED**

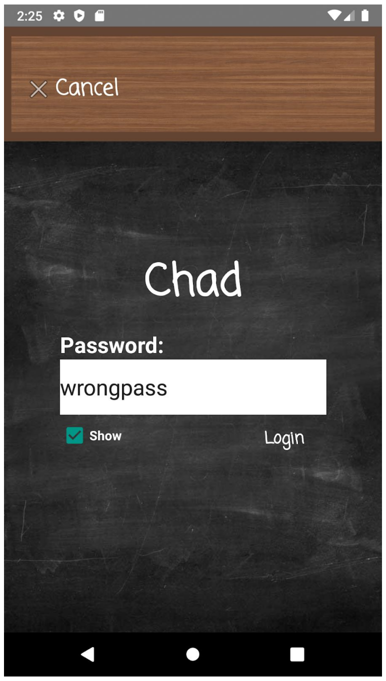


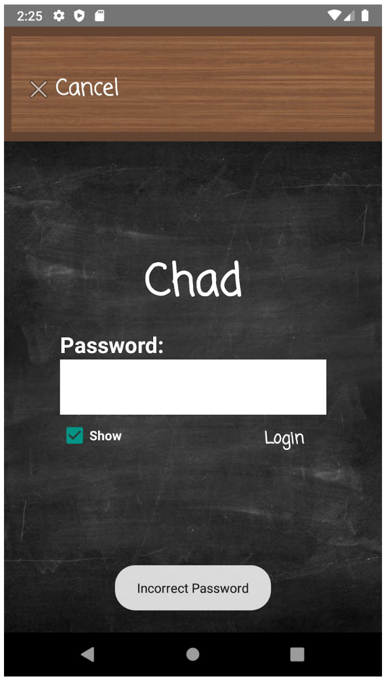
```
private void createUser(RegisterUserActivity mainActivity, String username, String password) {
    //User was accepted, create the user

    GameCategoryDBHelper gameCategoryDB = new GameCategoryDBHelper(mainActivity);
    GameTaskDBHelper gameTasksDB = new GameTaskDBHelper(mainActivity);
    gameTasksDB.initializeTaskObjects(username);
    RegisterUserDBHelper registerUserDB = new RegisterUserDBHelper(mainActivity);

    registerUserDB.addUser(username, password); ← Not encrypted

    gameCategoryDB.addCategory(username,  category: "Matching");

    gameCategoryDB.close();
    gameTasksDB.close();
    registerUserDB.close();
    return;
}
```

Secure Log-in: **PASS**

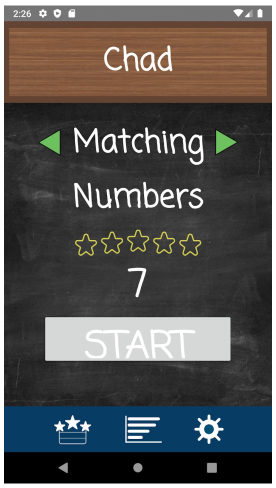

2:25
Cancel
Chad
Password:
Abc123@@
Show
Login

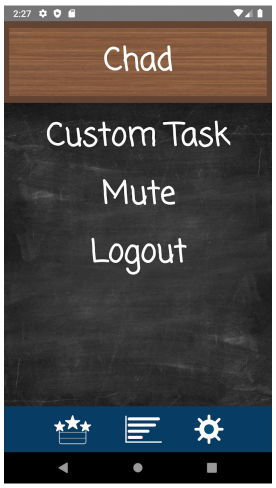

2:25
Chad
Matching
Numbers
7
START

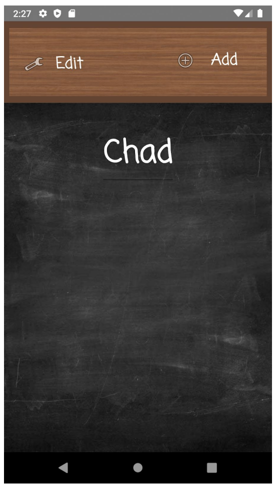

2:25
Cancel
Chad
Password:
wrongpass
Show
Login

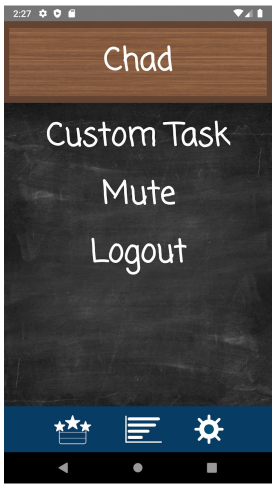

2:25
Cancel
Chad
Password:
Show
Login
Incorrect Password

Secure Log-out: **FAILED**

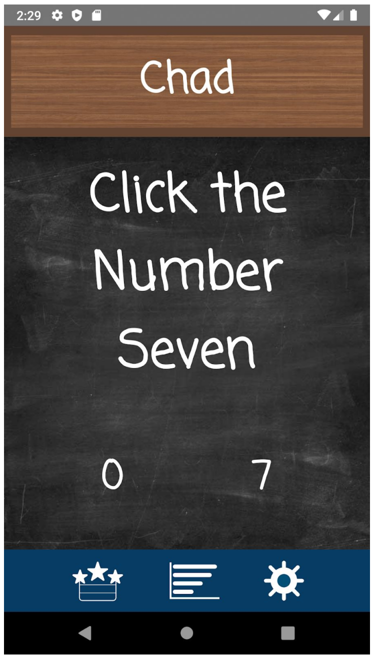


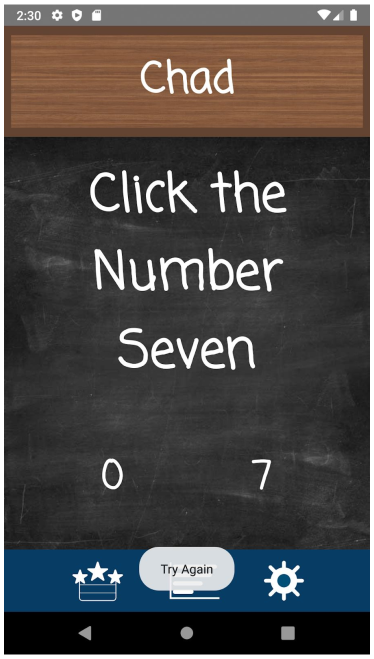


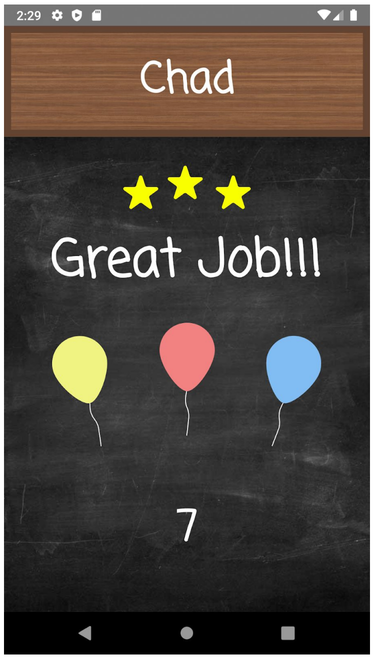


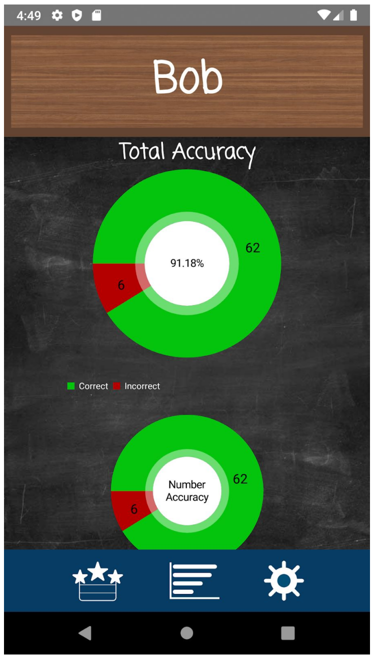


Functional Game: **PASS**

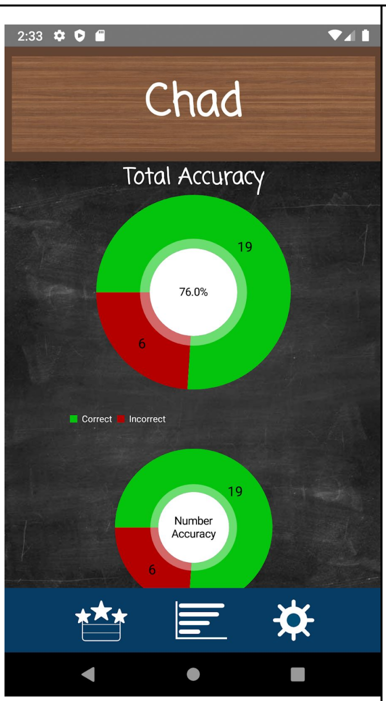


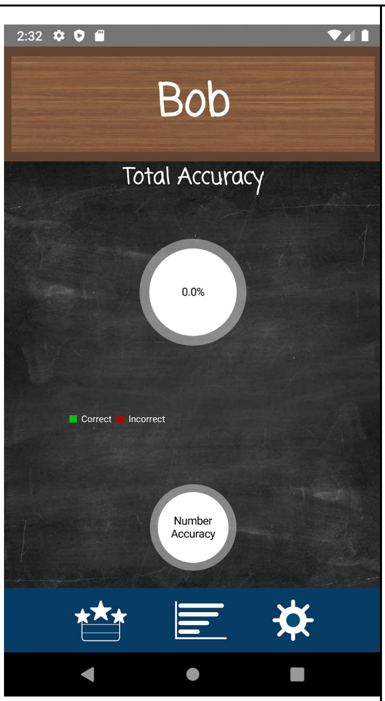


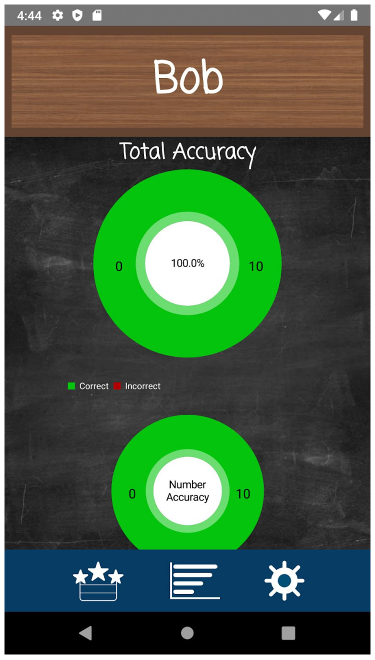




Visual Statistics: **PASS**


2:33
Chad
Total Accuracy
19
76.0%
6
Correct
Incorrect
19
Number Accuracy
6


2:32
Bob
Total Accuracy
0.0%
Correct
Incorrect
Number Accuracy


4:44
Bob
Total Accuracy
0
100.0%
10
Correct
Incorrect
0
Number Accuracy
10