CA326 Third Year Project - Testing and Validation

Squid Proxy for Safe Browsing
Completed: 23/02/23

Project Title: Squid proxy and Android app for safe browsing
Students: Aaron Crawford - 20336753
Ciaran Skelly - 20324213
Supervisor: Darragh O'Brien

1. Manual Testing

Going through each function of the app manually to test if the UI displays correctly and all functions work as intended. Raspberry pi was used to host the SSH server and squid proxy, android phone was used to use the app and change the configuration of the proxy server on the raspberry pi and a laptop running google chrome was the device used to test restrictions were working.

Page	**Function**	**Expected Result**	**Actual Result**	**Outcome**
Home	Selecting button	After selecting a button a user is taken to that buttons page	After selecting a button a user is taken to that buttons page	PASS
Home	Query Info	When on Home page the proxy's weekly query logs display at the bottom of the page	When on Home page the proxy's weekly query logs display at the bottom of the page	PASS
System Wide Blacklist	Blacklisted websites display	All current Blacklisted websites display on the page	All current Blacklisted websites display on the page	PASS
System Wide Blacklist	Adding a website to blacklist	Once add it pressed and a word is typed the word gets added to list below	Once add it pressed and a word is typed the word gets added to list below	PASS

Page	Function	Expected Result	Actual Result	Outcome
System Wide Blacklist	Pressing add when no word is typed	A red circle with an exclamation point appears as a warning. No other functions run	A red circle with an exclamation point appears as a warning. No other functions run	PASS
System Wide Blacklist	Adding a website to blacklist	Website gets sent to blocked sites file on raspberry pi	Website gets sent to blocked sites file on raspberry pi	PASS
System Wide Blacklist	Users can no longer access website	The proxy blocks the user from accessing the website	The proxy blocks the user from accessing the website	PASS
System Wide Blacklist	Clicking on word in website blacklist list	Word gets automatically added to text box below above button remove	Word gets automatically added to text box below above button remove	PASS
System Wide Blacklist	Remove when no website is selected	Pop appears informing user to select website first	Pop appears informing user to select website first	PASS
System Wide Blacklist	Removing a website	Website no longer appears in blocked website list above	Website no longer appears in blocked website list above	PASS
System Wide Blacklist	Removing a website	Website gets removed from blocked sites file on raspberry pi	Website gets removed from blocked sites file on raspberry pi	PASS
System Wide Blacklist	Removing a website	Users can now access website that was removed	Users can now access website that was removed	PASS
User Settings	List of users displays	All users that have been created are displayed	All users that have been created are displayed	PASS
User Settings	Pressing create user with no info supplied	A red circle with an exclamation point appears as a warning. Both password and username needed	A red circle with an exclamation point appears as a warning. Both password and username needed	PASS

Page	Function	Expected Result	Actual Result	Outcome
User Settings	Creating a user with a username and password	Username and password get added to the htpasswd file. Password becomes encrypted	Username and password get added to the htpasswd file. Password becomes encrypted	PASS
User Settings	Creating a user with a username and password	Necessary lines for created user get added to squid.conf file	Necessary lines for created user get added to squid.conf file	PASS
User Settings	Creating a user with a username and password	New user's name gets added to list on app's UI	New user's name gets added to list on app's UI	PASS
User Settings	Clicking on a username displayed	Brought to that user's individual settings page	Brought to that user's individual settings page	PASS
Individual User Settings	List of blocked websites for user	The list of websites blocked for this specific user show in a list	The list of websites blocked for this specific user show in a list	PASS
Individual User Settings	Clicking on text boxes to enter time	Pop up appears of clock for user to enter time	Pop up appears of clock for user to enter time	PASS
Individual User Settings	Clicking keyboard bottom left of clock pop up	Clock changes to text box where user can manually type in time	Clock changes to text box where user can manually type in time	PASS
Individual User Settings	Pressing ok when time is entered	Time entered is now displayed in text box for user to see	Time entered is now displayed in text box for user to see	PASS
Individual User Settings	Clicking Set Time Limit	Time gets sent to squid.conf file on raspberry pi	Time gets sent to squid.conf file on raspberry pi	PASS

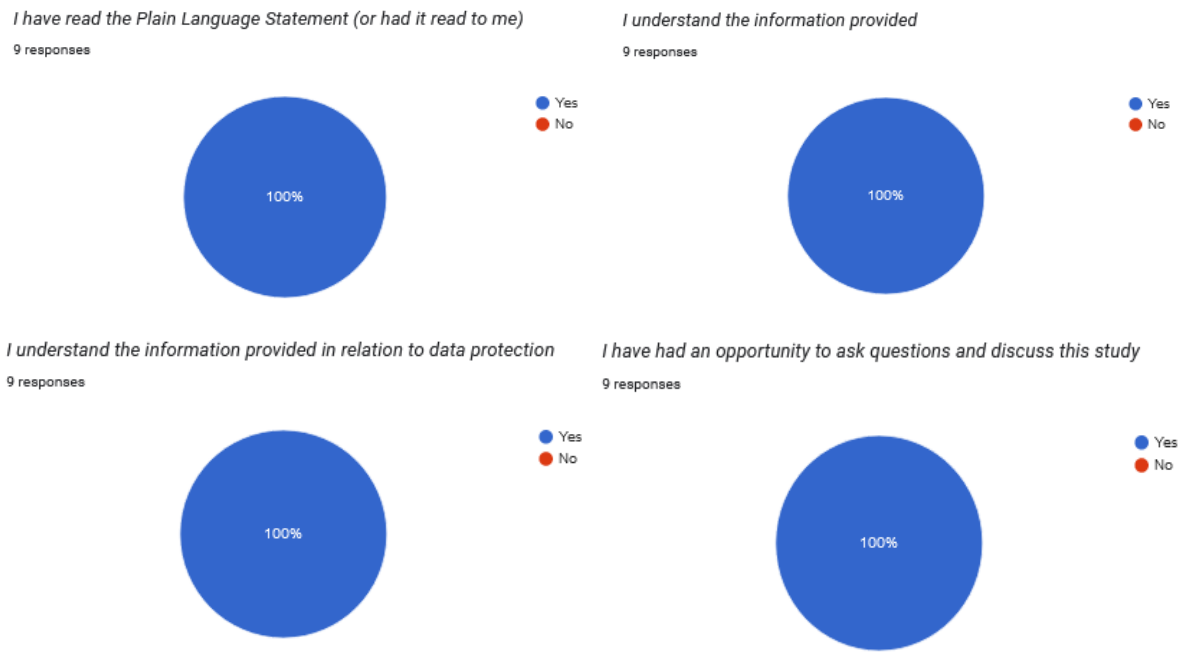
Page	Function	Expected Result	Actual Result	Outcome
Individual User Settings	Clicking Set Time Limit	User is now restricted to only being able to access the internet inside this time period	User is now restricted to only being able to access the internet inside this time period	PASS
Individual User Settings	Clicking Remove Limit	Squid.conf file gets reset to original settings	Squid.conf file gets reset to original settings	PASS
Individual User Settings	Clicking Remove Limit	User is no longer restricted by time constraints	User is no longer restricted by time constraints	PASS
Individual User Settings	Individual user blacklist	This blacklist works the same as network wide except only for specific user	This blacklist works the same as network wide except only for specific user	PASS
Individual User Settings	Turning on individual blacklist	On button becomes a darker shade and off button a lighter shade	On button becomes a darker shade and off button a lighter shade	PASS
Individual User Settings	Turning on individual blacklist	Squid.conf file updates to turn on this user blacklist	Squid.conf file updates to turn on this user blacklist	PASS
Individual User Settings	Turning off individual blacklist	Off button becomes a darker shade and on button a lighter shade	Off button becomes a darker shade and on button a lighter shade	PASS
Individual User Settings	Turning off individual blacklist	Squid.conf file updates to turn off this user blacklist	Squid.conf file updates to turn off this user blacklist	PASS
User Logs	On loading page	List of all users displays	List of all users displays	PASS
User Logs	Clicking on a username	This brings you to that individual user’s logs	This brings you to that individual user’s logs	PASS

Page	Function	Expected Result	Actual Result	Outcome
Individual User's Logs	On loading page	The list of the user's logs over the past week displays	The list of the user's logs over the past week displays	PASS
Individual User's Logs	Selecting view daily logs	The list of the user's logs over the past day displays	The list of the user's logs over the past day displays	PASS

2. User Testing

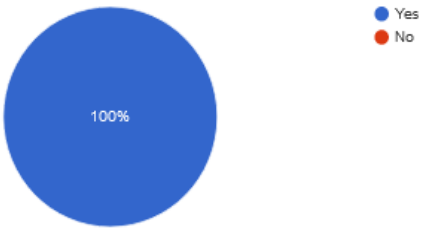
Nine CASE3 students were asked to use our application and complete a survey on their opinions of the UI and features of the application. Testing was conducted over two days the 21st and 22nd of february and surveys were emailed along with online consent forms to participants. Questions on the survey were based on information that we required and what we learnt from (Palmer, maze.co).

Consent Form Questions



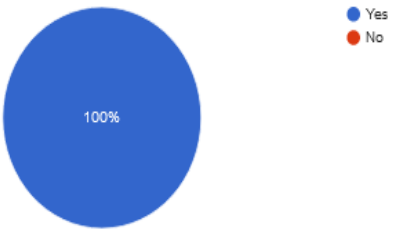
I have received satisfactory answers to all my questions

9 responses



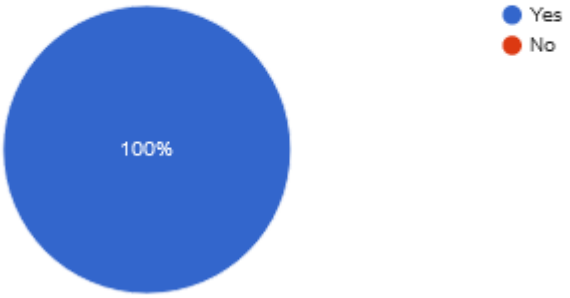
I understand I may withdraw from the Research study at any point

9 responses



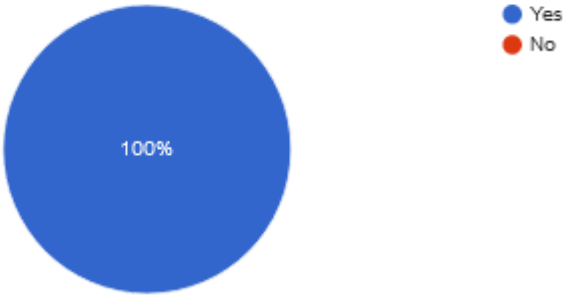
I have read and understand the arrangements to be made to protect confidentiality of information provided is subject to legal limitations

9 responses



I consent to participate in this research study

9 responses

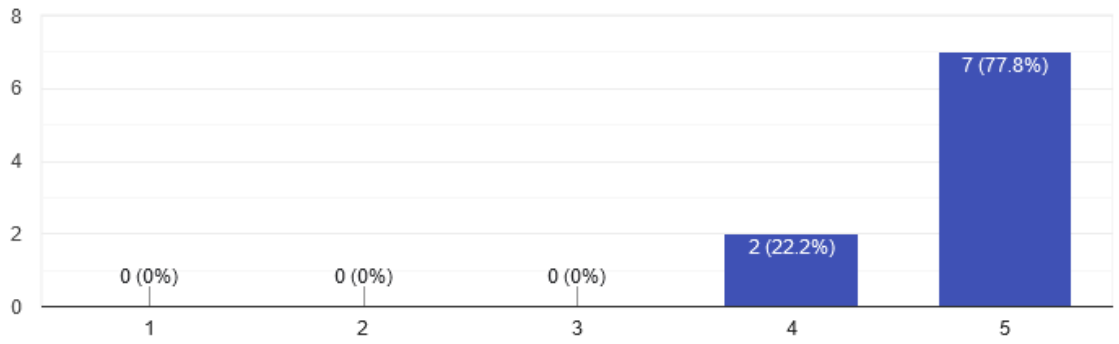


Survey Questions

How easy did you find navigating the UI

Copy

9 responses

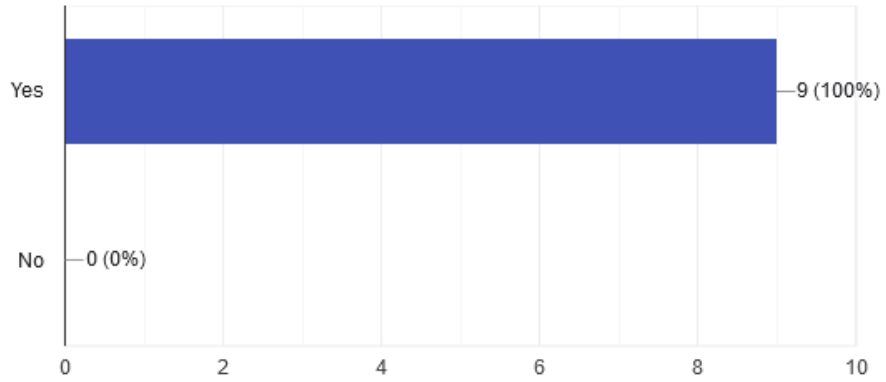


First we asked participants how easy they found navigating the app. This was important to us as many of our potential users may not be tech savvy so clear navigation was a must.

Was the language in the mobile app clear and straightforward?

Copy

9 responses

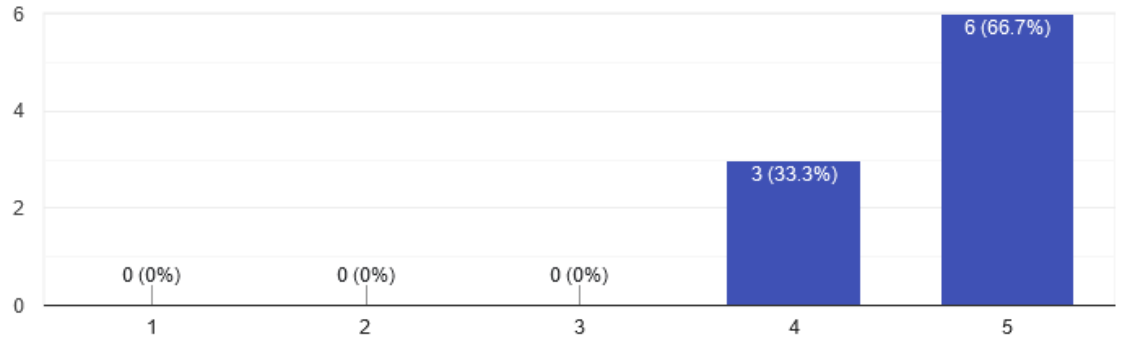


We then followed up by asking how clear the language we used throughout the app was. Similar to the last question it was important to us that everything was clearly labelled for users to understand.

How easy did you find it to use the time limits feature?

Copy

9 responses



We then asked users to rate how easy they found using the time limit feature which we deemed to be the feature that was the most confusing. However everyone participant found this easy to achieve and straightforward.

Did you find any of the features particularly hard to use and if yes explain why

9 responses

No

No, everything seemed to be easy to use

i found the features nice and clear to use

None

No, nothing was particularly difficult

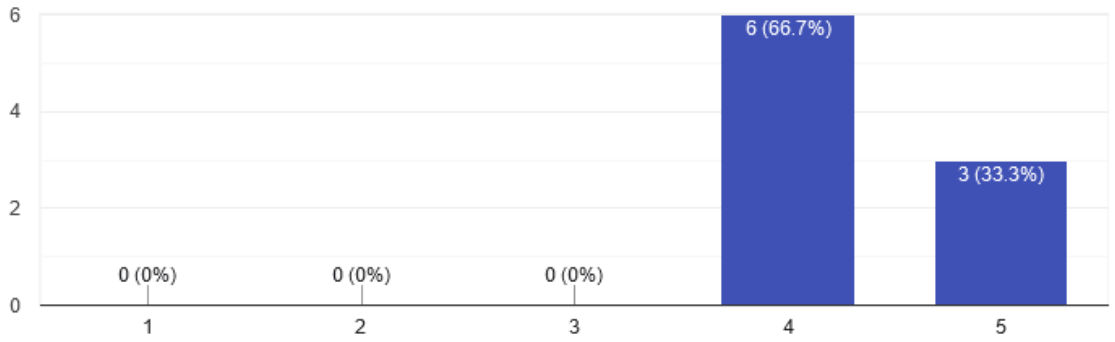
No, everything was clear

We then followed up the previous question by asking if there were any other features participants found difficult. We were glad to see all features were straight forward and not confusing for users.

How would you rate the overall performance of the app?

Copy

9 responses



Next we asked how they rated the performance of the app. Here 66% of participants said 4 which was related to the fact that it can take a second for the app to gather information off the files on the Raspberry pi, but still a high rating.

Are there any additional features you think are missing or could benefit the app?

9 responses

No

The idea is good, would be nice to have a web version too. Overall really cool product

No additional features I would like to see on this app

a whitelisting feature would be a nice addition

whitelisting

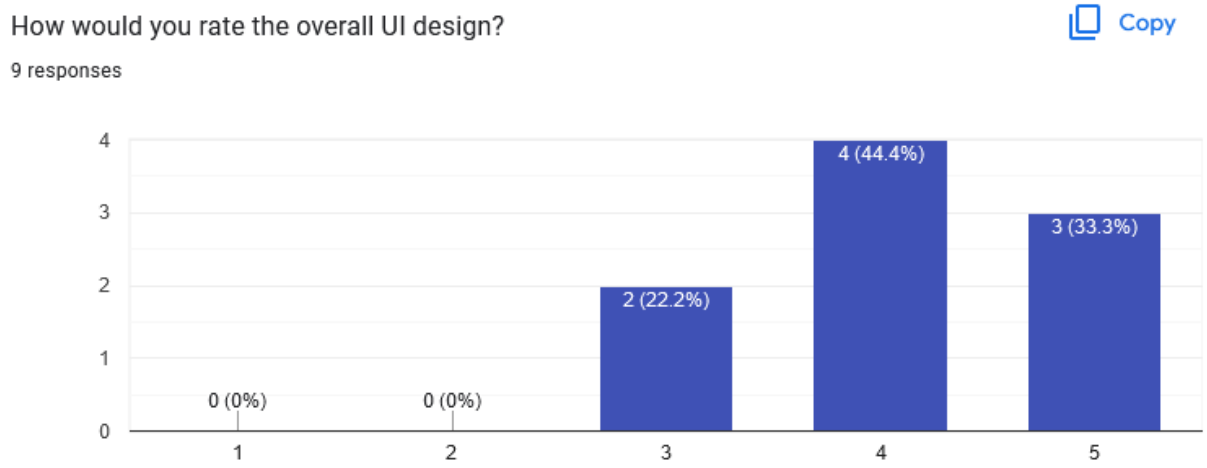
The UI could use a little work—why have 2 buttons to turn something on/off when one can do the same thing? (or use a switch).

Also, buttons that are destructive (cause something to get deleted) should ideally be red in colour, as users associate red with deleting or closing things (windows close button, iOS decline call button etc)

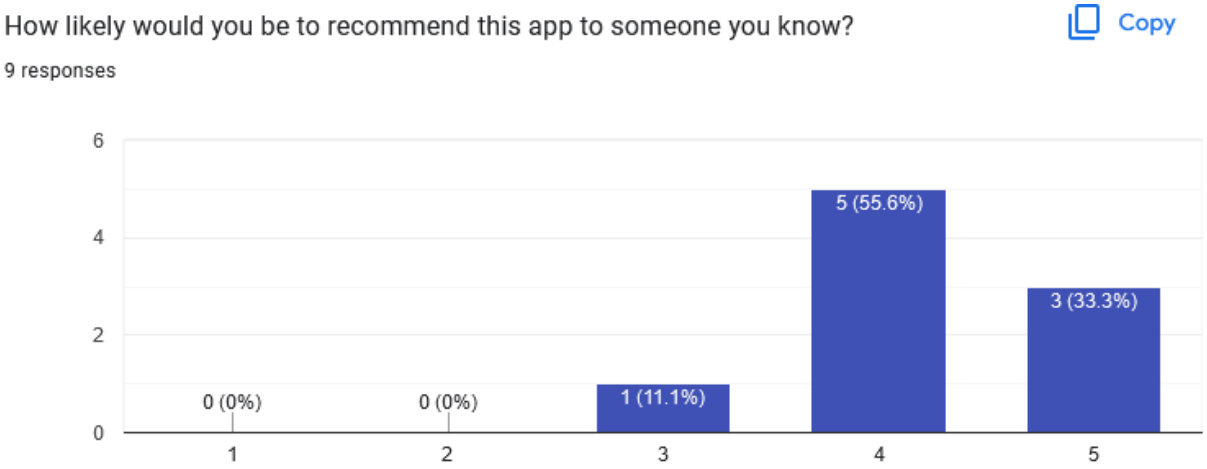
No additional features

We then asked for additional features they would like to see and two participants said whitelisting which was a feature we attempted to introduce however as websites for example youtube host some of their content such as images and videos on separate domains, We could know what additional websites we would have to allow so that this content would be allowed to be displayed properly. Another answer we got which we found great was based on our buttons for turning

blacklisting on and off. Before they both were identical and there was no way of telling if someone's blacklist was on or off. After this we changed it so that if the blacklist was on the on button was a darker shade and the off button was a lighter shade and vice versa. Finally a participant said that destructive buttons should be red which we found very useful and after the surveys we changed all remove or delete buttons red. We found great benefit out of this question in particular.



We averaged a score of 4 for the overall rating of our UI which we believe we improved on after the user testing.



Finally we asked how likely user were to recommend this application to others and got positive feedback that showed there was a use for our app.

3. Espresso Testing

We performed automated ui testing in android studio using espresso (developer.android.com). Espresso is an open-source

framework created by Google for Android that allows you to write tests on the user interface. It waits for asynchronous tasks to be completed before moving to the next test which we found very useful because our app has a number of asynchronous tasks that make changes to the ui, so it made testing these features much easier. While the tests were running we had the squid proxy running on the raspberry pi. Espresso testing allowed us to test the ui more efficiently than manual testing and check that there were no regressions in the code as we added new features or made changes.

Test 1:

Code:

```
@Test
public void titleIsAboveButtonBlacklist(){
    onView(withText("Safe Browsing")).check(matches(isDisplayed()));
    onView(withText("Safe Browsing")).check(isCompletelyAbove(withId(R.id.blacklist_btn)));
}
```

Description:
The first test we added was just used to learn how to create and run the tests. The test checks that the text “Safe Browsing” is displayed and that it is above the blacklist button.

Test 2:

Code:

```
@Test
public void blacklistActivityTest(){
    onView(withId(R.id.blacklist_btn)).perform(click());
    onView(withText("Blacklist")).check(matches(isDisplayed()));

    //checking adding website error messages
    onView(withId(R.id.add_btn)).perform(click());
    onView(withId(R.id.typed_website)).check(matches(hasErrorText("You must enter a website")));

    //checking adding website functionality
    onView(withId(R.id.typed_website)).perform(typeText("twitch"), ViewActions.closeSoftKeyboard());
    onView(withId(R.id.add_btn)).perform(click());
    onData(anything()).inAdapterView(withId(R.id.blacklisted)).atPosition(2).check(matches(withText("twitch")));

    //checking removing website functionality
    onData(anything()).inAdapterView(withId(R.id.blacklisted)).atPosition(2).perform(click());
    onView(withId(R.id.selected)).check(matches(withText("twitch")));
    onView(withId(R.id.remove_btn)).perform(click());
    onView(withId(R.id.selected)).check(matches(withHint("click a website to remove")));
}
```

Description:

The test blacklistActivityTest tests the ui of the Blacklist activity. The test clicks a button to open the blacklist activity and checks that the text “Blacklist” is displayed. The text input box error messages are then checked by clicking the add button before a website has been input. Adding a website to the blacklist is checked by typing in a new website, clicking the add button and then checking data at the position where it should have been added. Removing is checked by clicking on an entry in the list, checking that the selected website text box has changed to display the selected website, then clicking the remove button and checking the selected website text box has changed back to just display the hint.

Test 3:

Code:

```
@Test
public void userSettingsActivityTest(){
    onView(withId(R.id.devices_btn)).perform(click());
    onView(withText("Select a user to change settings")).check(matches(isDisplayed()));

    //checking selecting a user to view
    onData(anything()).inAdapterView(withId(R.id.connected)).atPosition(0).perform(click());
    onView(withId(R.id.page_title)).check(matches(withText("testing123")));
    onView(withId(R.id.page_title)).check(isCompletelyAbove(withText("Individual Time Limit Options")));

    //checking the times are put into the text field
    onView(withId(R.id.time_1)).check(matches(withHint("Start Time")));
    onView(withId(R.id.time_1)).perform(click());
    onView(withText("OK")).perform(click());
    onView(withId(R.id.time_1)).check(matches(withText("00:00")));
    onView(withId(R.id.time_2)).check(matches(withHint("End Time")));
    onView(withId(R.id.time_2)).perform(click());
    onView(withText("OK")).perform(click());
    onView(withId(R.id.time_2)).check(matches(withText("00:00")));

    //checking adding website error messages
    onView(withId(R.id.add_btn)).perform(click());
    onView(withId(R.id.typed_website)).check(matches(hasErrorText("You must enter a website")));

    //checking adding website functionality
    onView(withId(R.id.typed_website)).perform(typeText("twitter"), ViewActions.closeSoftKeyboard());
    onView(withId(R.id.add_btn)).perform(click());
    onData(anything()).inAdapterView(withId(R.id.whitelist_sites)).atPosition(2).check(matches(withText("twitter")));

    //checking removing website functionality
    onData(anything()).inAdapterView(withId(R.id.whitelist_sites)).atPosition(2).perform(click());
    onView(withId(R.id.selected)).check(matches(withText("twitter")));
    onView(withId(R.id.remove_btn)).perform(click());
    onView(withId(R.id.selected)).check(matches(withHint("click a website to remove")));
}
```

Description:

The test userSettingsActivityTest tests the ui of the individual users settings activity. The test clicks a button to open the activity that lists all users and checks that the text “Select a user to change settings” is displayed. A user is clicked and the individual user activity is started, the page title is checked for its text and position, then setting the time limit start and end time are checked by opening the time picker dialog, clicking ok and then checking that the start and end time text boxes changed.

Adding and removing are checked using the same process as the blacklist page described previously.

Test 4:

Code:

```
@Test
public void userLogsActivityTesting() {
    onView(withId(R.id.user_logs_btn)).perform(click());
    onView(withText("Users")).check(matches(isDisplayed()));
    onView(withText("Users")).check(isCompletelyAbove(withText("Select a User to view their logs")));

    //checking selecting a user to view
    onData(anything()).inAdapterView(withId(R.id.connected)).atPosition(0).perform(click());
    onView(withId(R.id.page_title)).check(matches(isDisplayed()));
    onView(withId(R.id.page_title)).check(isCompletelyAbove(withId(R.id.weekly_check)));

    //check the checkboxes
    onView(withId(R.id.weekly_check)).check(matches((isChecked())));
    onView(withId(R.id.daily_check)).check(matches((isNotChecked())));
    onView(withId(R.id.daily_check)).perform(click());
    onView(withId(R.id.daily_check)).check(matches((isChecked())));
    onView(withId(R.id.weekly_check)).check(matches((isNotChecked())));
    onView(withId(R.id.daily_check)).perform(click());
    onView(withId(R.id.daily_check)).check(matches((isChecked())));
    onView(withId(R.id.weekly_check)).check(matches((isNotChecked())));
}
```

Description:

The test userLogsActivityTesting tests the ui of the user logs activity. The test clicks a button to open the activity to select a user to read logs and checks that the text “Users” is displayed and is above another text element. A user is selected from the list by clicking the position of an item which starts the view logs activity, the title is checked if it is displayed and for its position in relation to the “weekly_check” checkbox. The checkboxes should always have one checked and one not checked so they are tested to see if they default to having weekly checked and daily not checked, then they are tested to make sure checking one box makes the other not checked, and finally they are tested to make sure that when a checkbox is checked it cannot be clicked again to have no checkbox checked.

Test 5:

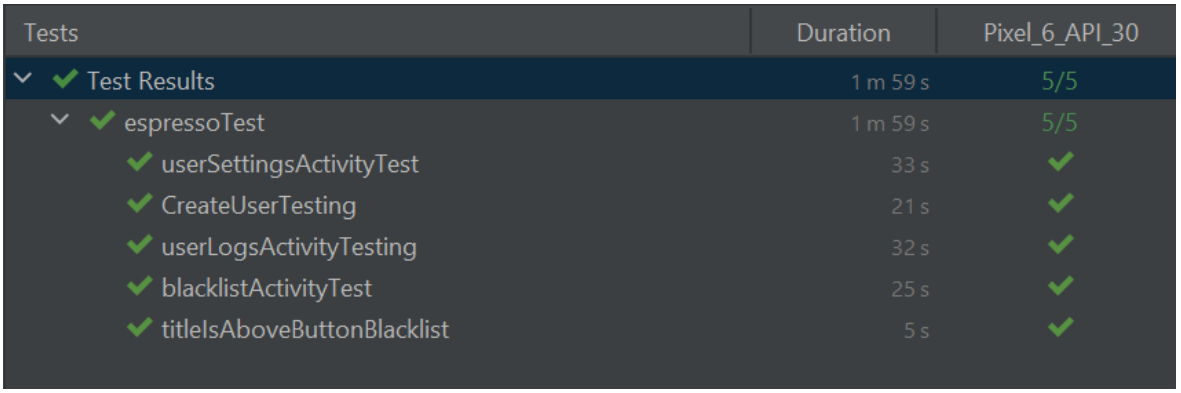
Code:

```
@Test
public void CreateUserTesting(){
    onView(withId(R.id.devices_btn)).perform(click());
    onView(withText("Select a user to change settings")).check(matches(isDisplayed()));

    //enter information
    onView(withId(R.id.username)).perform(typeText("Bob"), ViewActions.closeSoftKeyboard());
    onView(withId(R.id.password)).perform(typeText("test123"), ViewActions.closeSoftKeyboard());
    onView(withId(R.id.create_user_btn)).perform(click());

    //check if user is created
    onData(anything()).inAdapterView(withId(R.id.connected)).atPosition(1).check(matches(withText("Bob")));
}
```

Description:https://developer.android.com/training/testing/espresso
The test CreateUserTesting tests the creating a user feature. The test clicks a button to open the activity that lists all users and checks that the text “Select a user to change settings” is displayed, a sample username is typed into the username edit text box and then a password is typed into the password edit text box, the create user button is clicked and then the list of users data is checked at the position the new username should have been added.

Results:

These tests were run on the final version of the ui and all passed. The results are detailed below with the time it took for each to finish.



References:

Survey questions:

Palmer, Taylor. “Usability Testing Questions: How to Ask Insightful UX Questions.” *Maze*, https://maze.co/guides/usability-testing/questions/ .

Espresso:

“Test in Android Studio.” *Android Developers*, 18 October 2022, https://developer.android.com/studio/test/test-in-android-studio.
“Espresso.” *Android Developers*, 27 October 2021, https://developer.android.com/training/testing/espresso.