NAME: SAKSHI VINAYAK KITTURKAR
CSU ID: 2860273
HOMEWORK 7

Make the following modification to the Menu app we just build:
Add a EditText for the user to enter a URL.
Place the button on the right of the EditText and rename to “submit”.
Add another button beneath the EditText and name it to” bookmark”.
On clicking the submit button, the URL should be displayed in the WebView.
On clicking the bookmark button, the current URL should be saved as a bookmark.
The menu for the app now should display a list of saved bookmarks.
On selecting a saved bookmark, the corresponding webpage should be displayed.

= In this menu app we can find a space where we must enter the URL and submit it and we have a button to make it a bookmark as well after that we have a list of all the URL that have been bookmarked.

In layout: MainActivity.java the application allows users to enter a URL in an EditText field, navigate to the entered URL, bookmark the current URL, and view and navigate to bookmarked URLs through a context menu and options menu.

These methods handle the selection of a bookmarked URL from the options menu (onOptionsItemSelected). The menuChoice method loads the selected URL into the WebView.

```
 @Override
public boolean onOptionsItemSelected(MenuItem item) {
    return menuChoice(item);
}

private boolean menuChoice(MenuItem item) {
    wv.loadUrl(bookmarks.get(item.getItemId()));
    return false;
}
```

These methods handle the creation of the context menu (onCreateContextMenu) and options menu (onCreateOptionsMenu). The createMenu method adds bookmarked URLs as menu items with their index as the item ID.

```
private void createMenu(Menu menu) {
    for (int i = 0; i < bookmarks.size(); i++) {
        MenuItem item = menu.add(0, i, i, bookmarks.get(i));
    }
}
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, view, menuInfo);
    createMenu(menu);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    createMenu(menu);
    return true;
}
```

In layout: ExampleInstrumentedTest.java this instrumented test checks whether the application context of the Android app under test has the expected package name, and it is executed on an Android device using JUnit 4 and the AndroidJUnit4 test runner.

The @Test annotation marks this method as a test case.The useAppContext method retrieves the application context of the app under test using InstrumentationRegistry.getInstrumentation().getTargetContext(). The assertEquals method checks whether the package name of the app matches the expected value (com.example.hw7).

```
@Test
public void useAppContext() {
    // Context of the app under test.
    Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
    assertEquals("com.example.hw7", appContext.getPackageName());
}
```

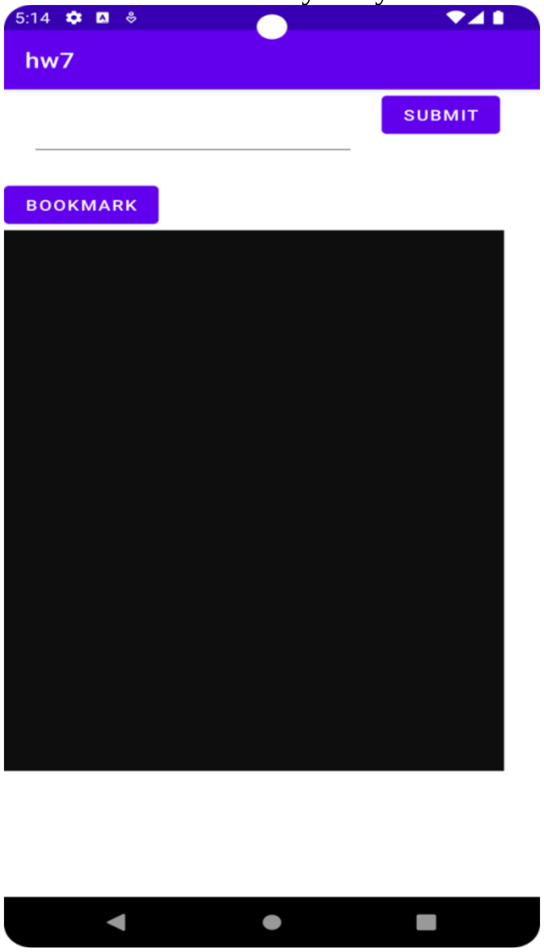
In layout: ExampleUnitTest.java this unit test checks whether the addition operation in the code (2 + 2) produces the expected result (4). It's a simple example of a local unit test that can be executed on the development machine during the software development process.

The @Test annotation marks this method as a test case. The addition_isCorrect method tests whether the addition operation 2 + 2 equals the expected result, which is 4. The assertEquals method checks whether the actual result (the sum of 2 + 2) matches the expected result (4).

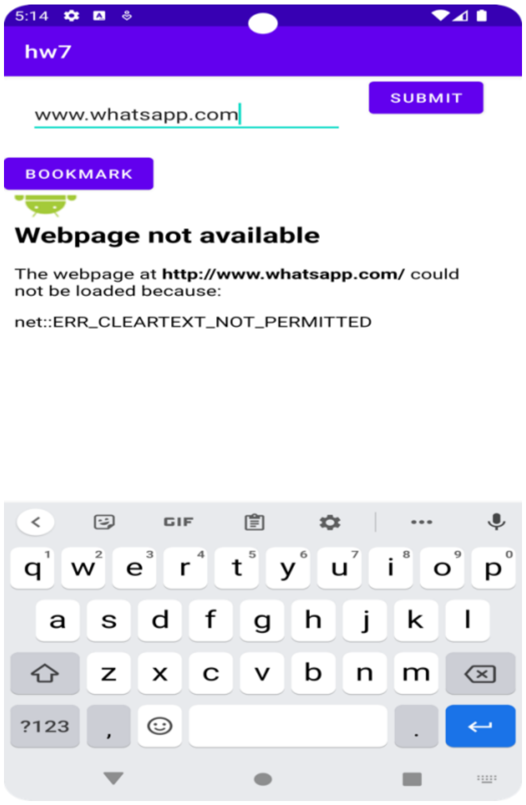
```
@Test
public void addition_isCorrect() {
    assertEquals(4, 2 + 2);
}
```

This assertion checks that the value of the expression 2 + 2 is equal to 4.
assertEquals(4, 2 + 2);

First Screen without any entry



Second screen with an submit and bookmark



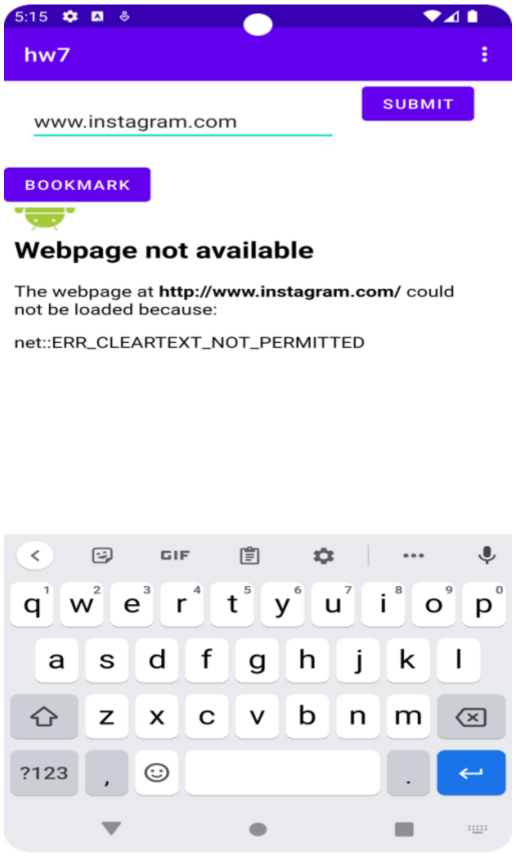
www.whatsapp.com

Webpage not available

The webpage at **http://www.whatsapp.com/** could not be loaded because:

net::ERR_CLEARTEXT_NOT_PERMITTED

Third screen with submit and bookmark



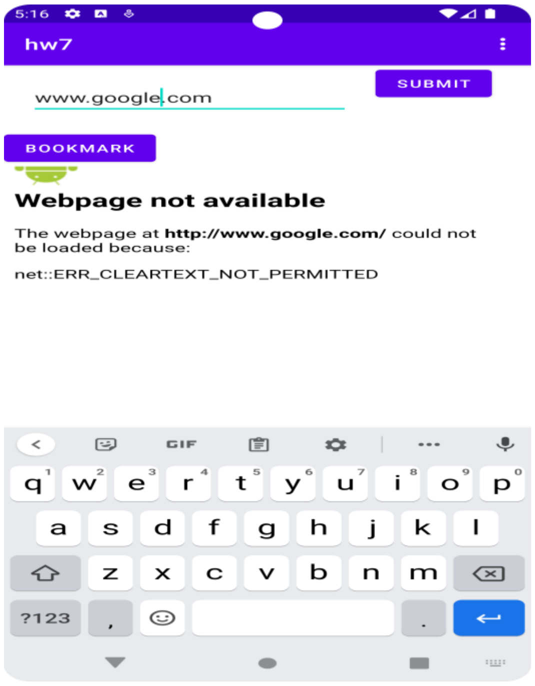
www.instagram.com

Webpage not available

The webpage at **http://www.instagram.com/** could not be loaded because:

net::ERR_CLEARTEXT_NOT_PERMITTED

Fourth screen with submit and bookmark



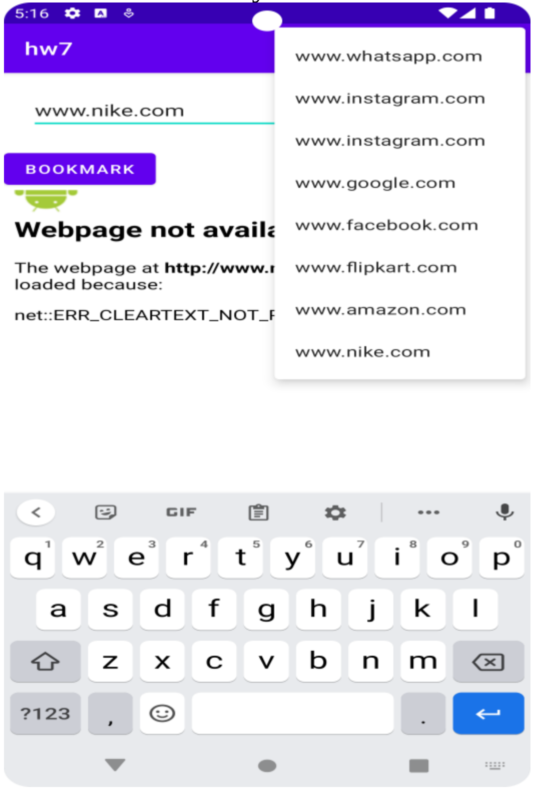
www.google.com

Webpage not available

The webpage at **http://www.google.com/** could not be loaded because:

net::ERR_CLEARTEXT_NOT_PERMITTED

Fifth screen shows you a list of all the bookmarked URL



www.nike.com

Webpage not availa

The webpage at **http://www.** loaded because:

net::ERR_CLEARTEXT_NOT_P

www.whatsapp.com
www.instagram.com
www.instagram.com
www.google.com
www.facebook.com
www.flipkart.com
www.amazon.com
www.nike.com