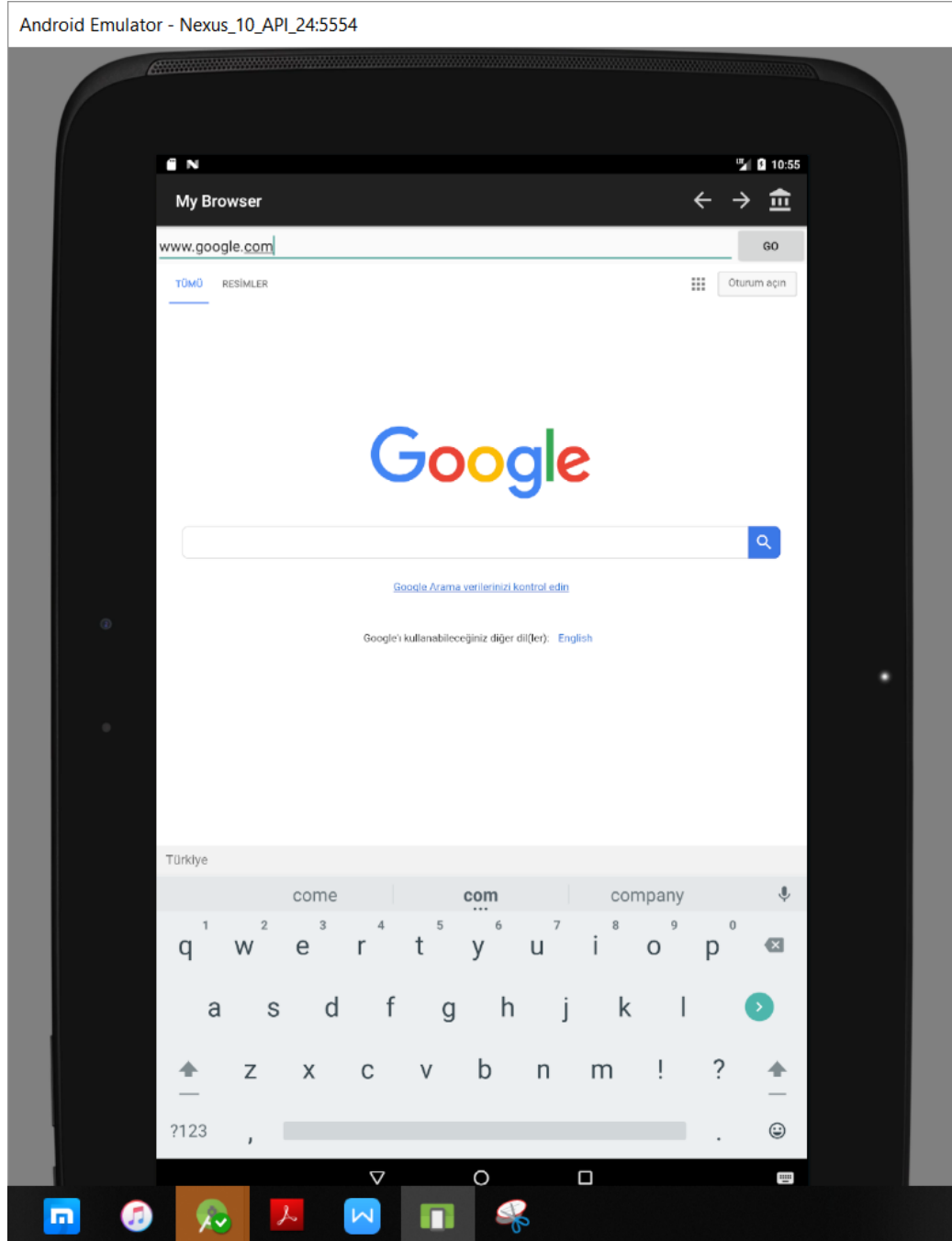
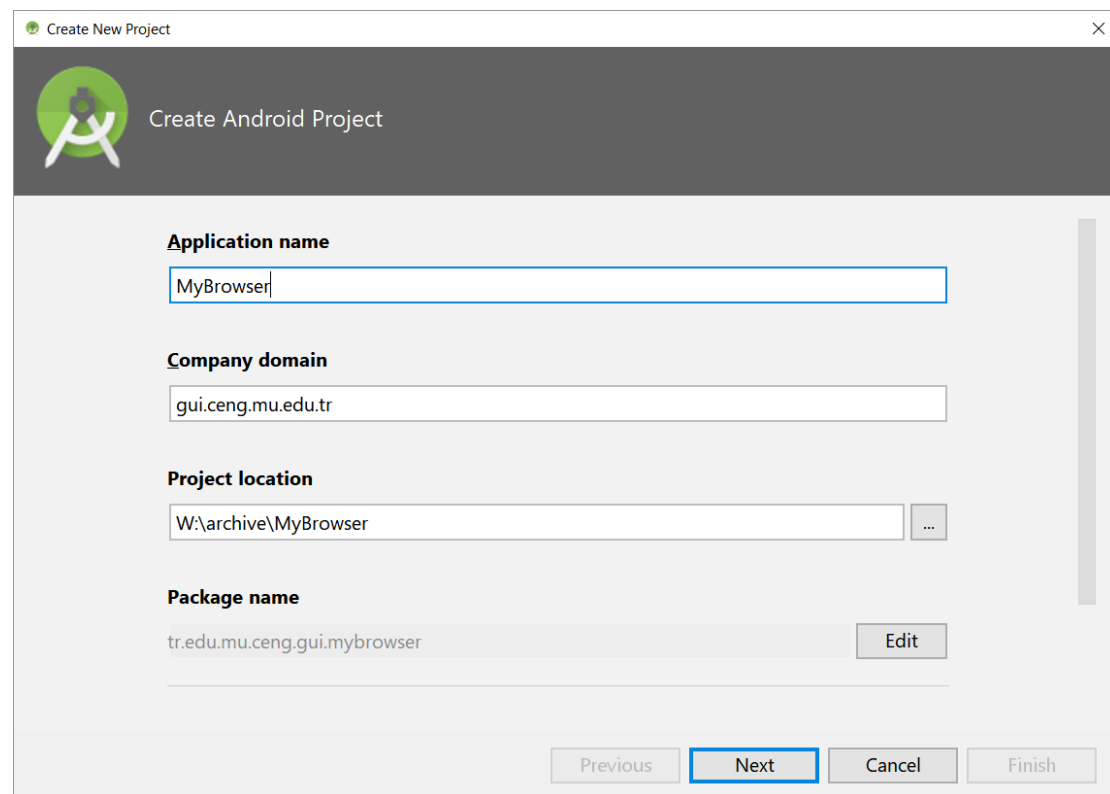


Lab 4: My Browser

In this lab we will be implementing the our own browser and register it as a web browser through intent filters.



1. Create new project in Android Studio.



The screenshot shows the 'Create New Project' dialog in Android Studio. The dialog has a title bar with the Android logo and the text 'Create New Project'. Below the title bar is a header area with the Android logo and the text 'Create Android Project'. The main area contains four sections: 'Application name' with a text field containing 'MyBrowser', 'Company domain' with a text field containing 'gui.ceng.mu.edu.tr', 'Project location' with a text field containing 'W:\archive\MyBrowser' and a browse button (...), and 'Package name' with a text field containing 'tr.edu.mu.ceng.gui.mybrowser' and an 'Edit' button. At the bottom, there are four buttons: 'Previous', 'Next' (highlighted with a blue border), 'Cancel', and 'Finish'.

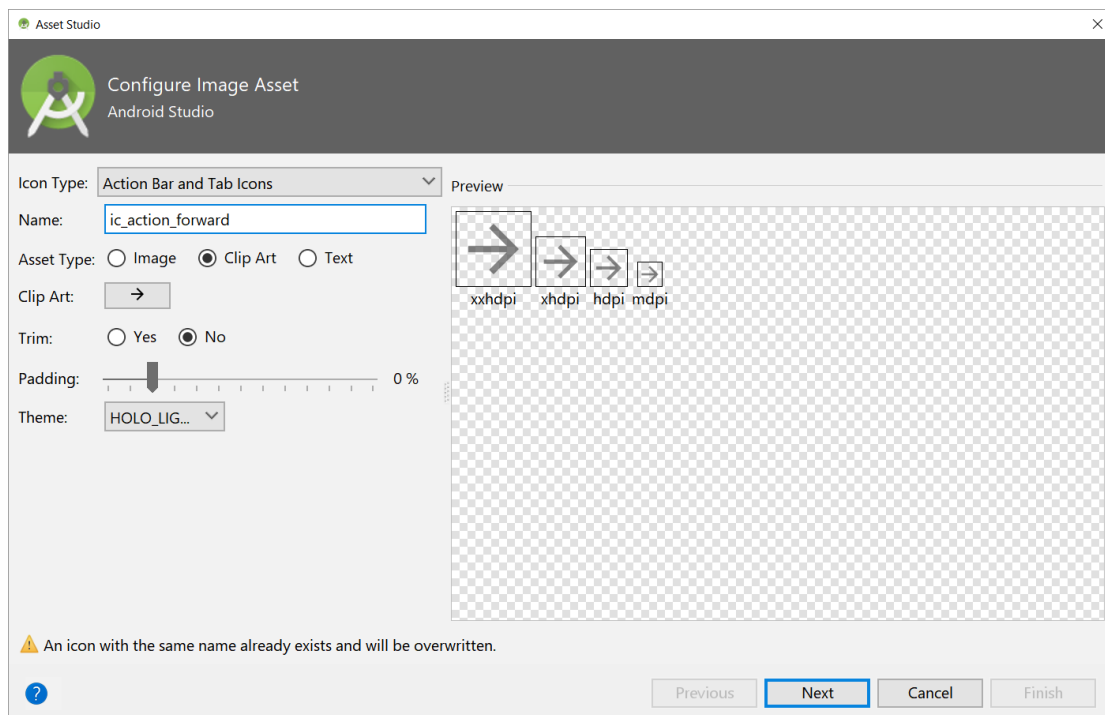
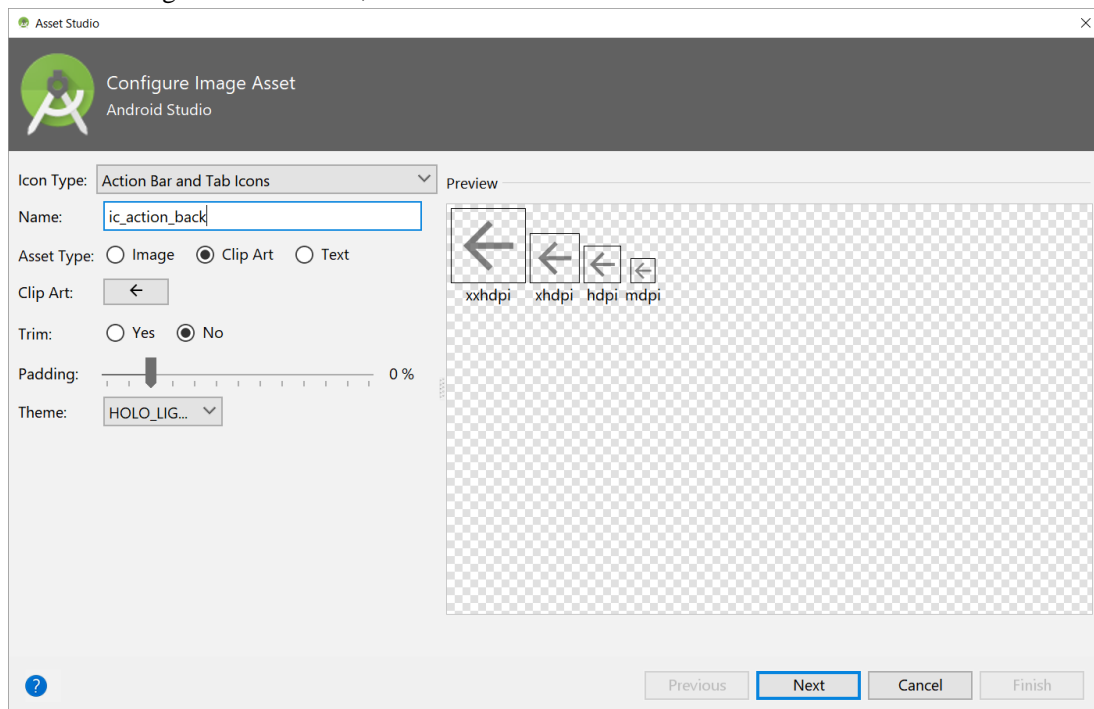
2. Select defaults for minimum SDK
3. Select Empty activity as the main activity
4. Add the following permissions to application.

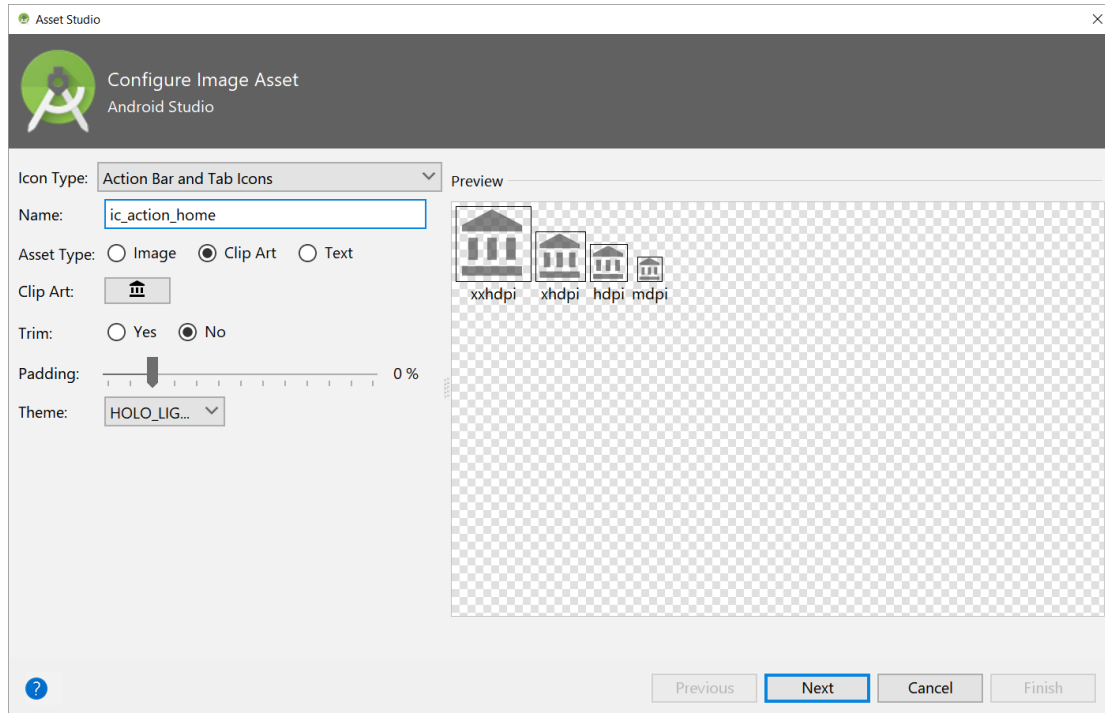
```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

5. Also set the android:usesCleartextTraffic attribute of application to true as shown below.

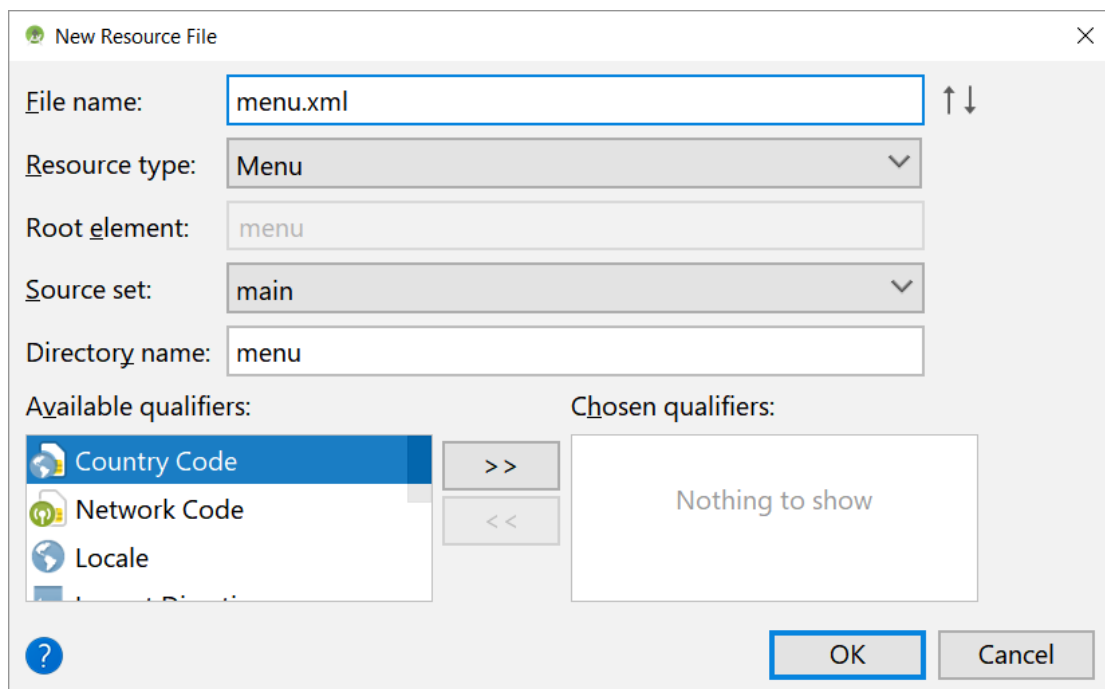
```
<?xml version="1.0" encoding="utf-8"?>
<manifest ...>
  <uses-permission android:name="android.permission.INTERNET" />
  <application
    ...
    android:usesCleartextTraffic="true"
    ...>
    ...
  </application>
</manifest>
```

6. Add image assets for back, forward and home





7. Create a new menu resource file



8. Modify the menu.xml as follows:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

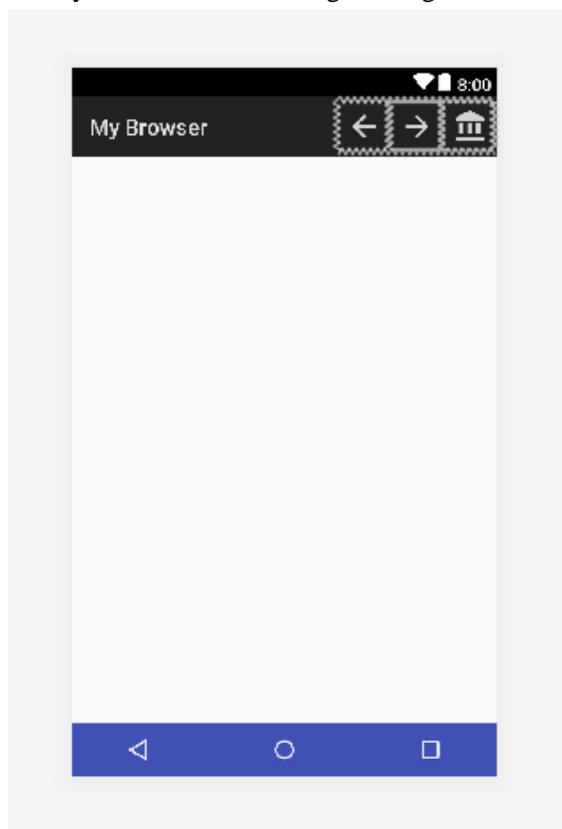
    <item
        android:id="@+id/item_back"
        android:enabled="false"
        android:icon="@drawable/ic_action_back"
        android:showAsAction="always"
        android:title="Back" />

    <item
        android:id="@+id/item_forward"
        android:icon="@drawable/ic_action_forward"
        android:showAsAction="always"
        android:title="Forward" />

    <item
        android:id="@+id/item_home"
        android:icon="@drawable/ic_action_home"
        android:showAsAction="always"
        android:title="Home" />

</menu>
```

Then you will have following in design view of menu.xml



9. Add the following method to the main activity to attach menu

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return super.onCreateOptionsMenu(menu);
}
```

10. Open the activity_main.xml and modify it as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/address"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="web address"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toStartOf="@+id/go"
        app:layout_constraintBottom_toBottomOf="@+id/go"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/go"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/address"
        app:layout_constraintTop_toTopOf="parent" />

    <WebView
        android:id="@+id/webview"
        android:layout_width="0dp"
```

```

        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/address" />
</android.support.constraint.ConstraintLayout>

```

11. Add the following statements to onCreate:

```

webView.setWebViewClient(new WebViewClient());

btnGo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        webView.loadUrl("http://" + txtAddress.getText());
    }
});

```

12. Register your browser in manifest file:

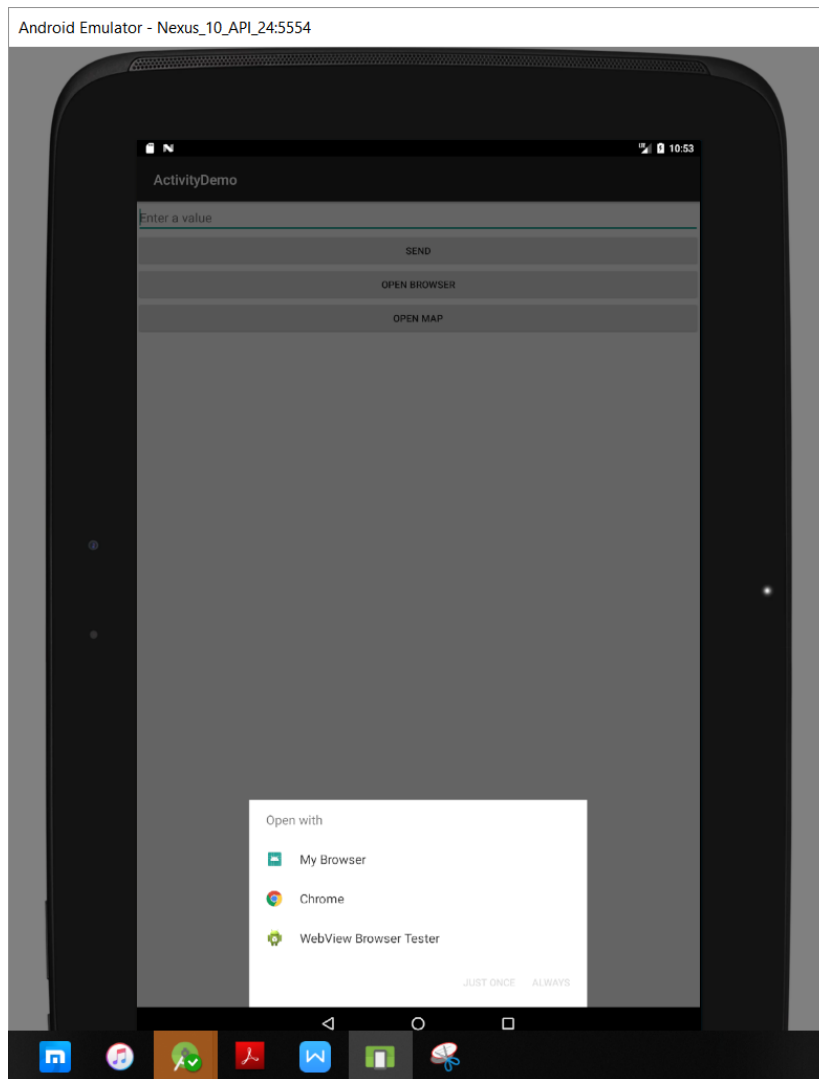
```

<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.MAIN" />

    <!-- Include the host attribute if you want your app to respond
         only to URLs with your app's domain. -->
    <data android:scheme="http" />
    <category android:name="android.intent.category.LAUNCHER" />
    <!-- The BROWSABLE category is required to get links from web pages. -->
    <category android:name="android.intent.category.BROWSABLE" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>

```

13. Start an activity with implicit intent to open a URL from another application



14. Add the following statements to onCreate of MyBrowser's main activity and repeat the 13th step:

```
if (getIntent() != null && getIntent().getData() != null) {  
    txtAddress.setText(getIntent().getData().toString());  
    webview.loadUrl(getIntent().getData().toString());  
}
```