
GitHub Username: nsanghi

Udacity Course Catalog

Description

Udacity has a long list of courses. While browsing the courses, one at times want to select certain courses to help filter down the interesting courses and make the description available offline. You may also want to share the interesting courses with your friends as well as setup a reminder to enroll in the course on a specific date.

This App provides a list of available Udacity Courses. The user can dive into the details of a course from the catalog and mark it favorite to make it available offline. The user can also share the course detail with others as well as create a reminder to enroll in the course.

Intended User

The main users of this app will be online learners who are interested to explore courses being offered by Udacity and create a bucket list with reminders to enroll in future.

Features

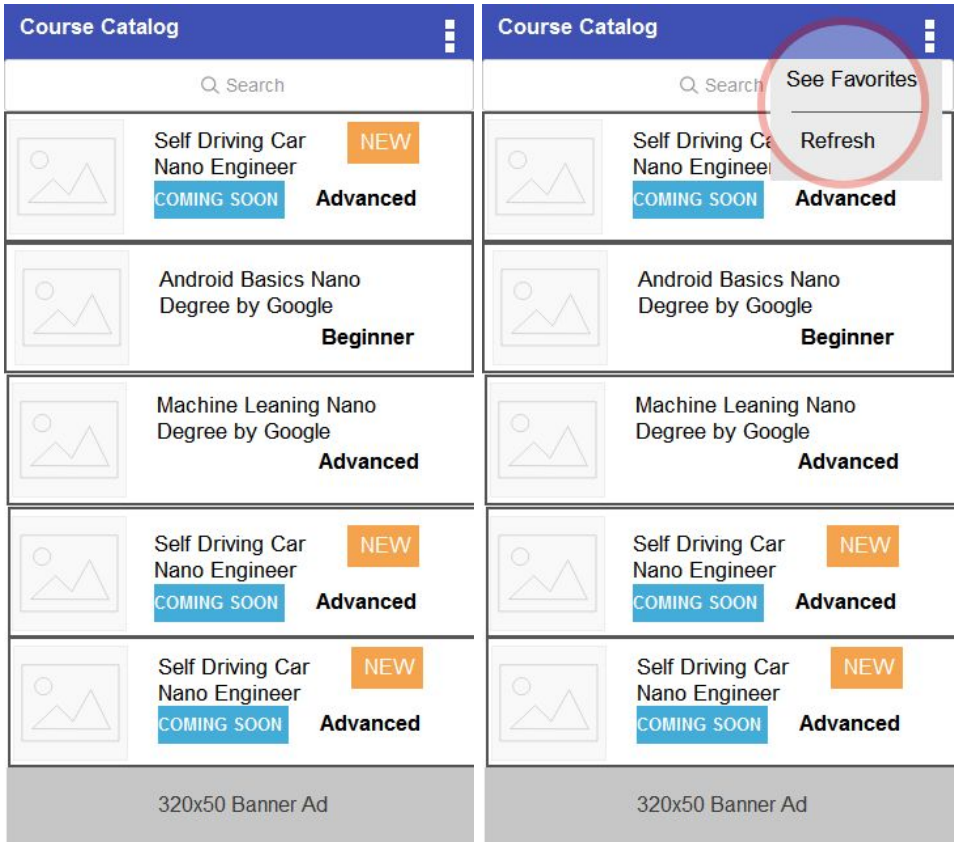
The app will:

- Upon launch, will present the user with a list of courses offered by Udacity
- Allows the user to filter the catalog using a search box
- Allows the user to view his favorite courses
- Allows the user to refresh the catalog on demand by a refresh action in the menu. The course catalog is also auto refreshed once a week.
- Allows the user to read the details of course by clicking on the course in the list
- Allows the user to mark a course as a favorite
- Allows the user to share a course url with others.
- Allows the user to open the Udacity course page in Browser
- Allows the user to view preview videos if available
- Allows the user to create a reminder (possibly for future enrollment) - OPTIONAL

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

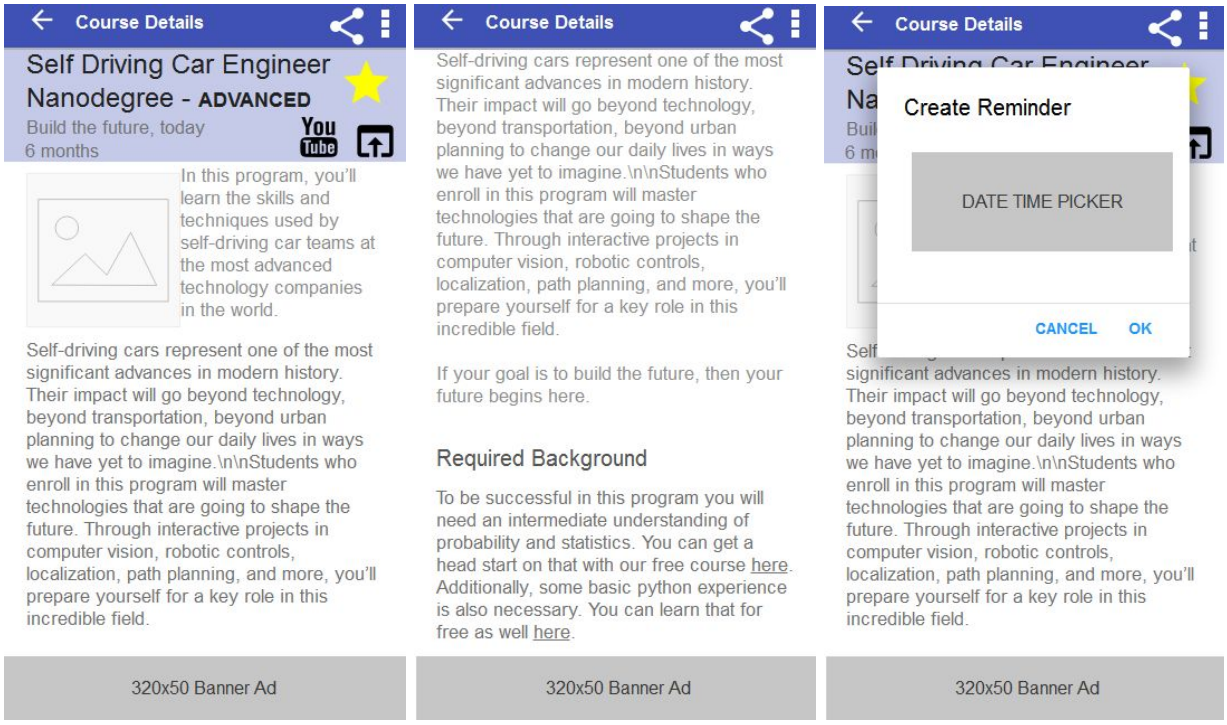


This is the Main screen that the user sees. It shows the Udacity Course Catalog with a search box at the top. Tapping any item will open Course Details Page.

From Actionbar Menu, you can filter by favorites or refresh the catalog.

There will be another copy of the layout without the banner ad for the paid flavour of the app.

Screen 2

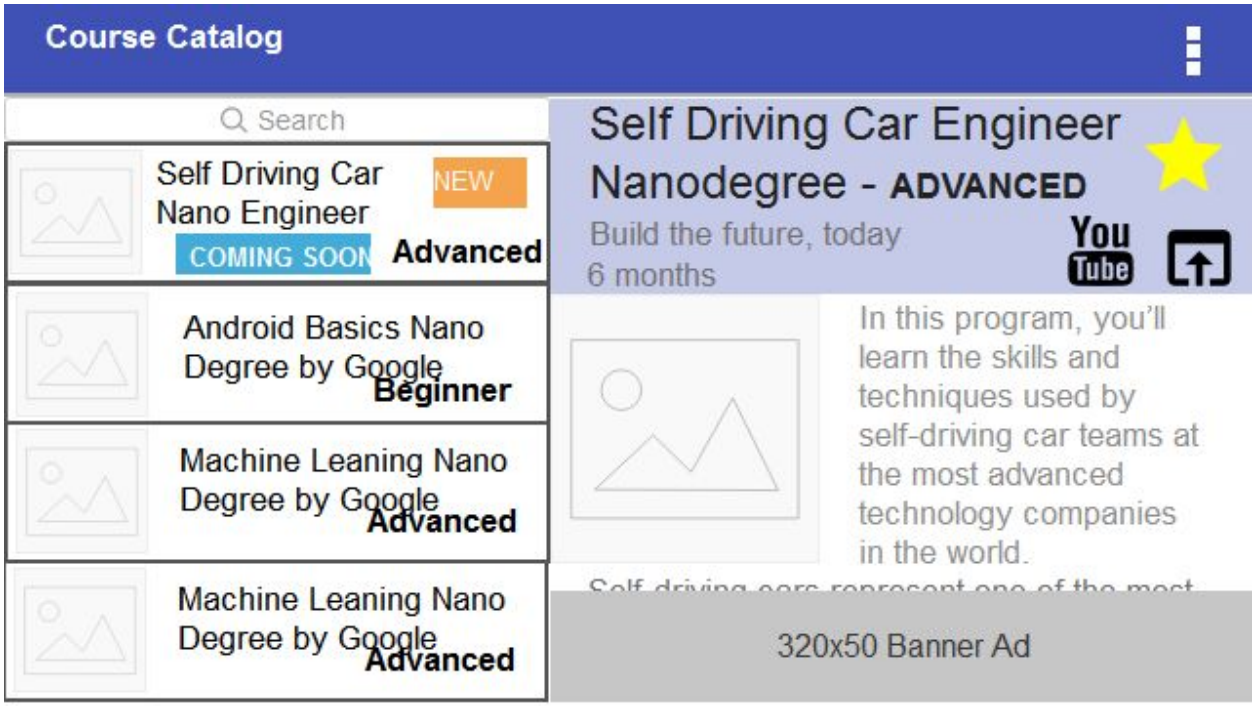


This is the Course Details Screen, It will show additional details of the course. From this screen, the user can preview course videos, if any. User can also see the udacity course page in Browser. The Star will allow the user to mark this course as favorite.

From Actionbar menu, the user will be able to share the course url and also create a reminder in his calendar.

There will be another copy of the layout without the banner ad for the paid flavour of the app.

Screen 3



This is the Tablet View. There will be another copy of the layout without the banner ad for the paid flavour of the app.

Key Considerations

How will your app handle data persistence?

The data will be persisted in local SQLLite database using Content Provider.

Describe any corner cases in the UX.

Once the User navigates out of the app to open URL in Browser or preview course video, the current activity will be put on backstack so that pressing the device back button will bring the user back.

Describe any libraries you'll be using and share your reasoning for including them.

I intend using Picasso to load the course Images.

Describe how you will implement Google Play Services.

I also intend to add Google Analytics and Google Admob to collect app usage and provide banner ads. These will be implemented using Firebase as per the setup explained above.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

The project will be built using Android Studio.

We will be using Picasso and Google Play Services.

- In the app's build.gradle, add mavenCentral under repositories section

```
repositories {
      mavenCentral()
}
```

- Next add **`compile 'com.squareup.picasso:picasso:2.5.2'`** to dependencies block.
- We also need to enable Google Analytics and Google Admob using Firebase. Details of the steps below is given at url https://firebase.google.com/docs/android/setup
    - Create a new project in Firebase Console
    - Download and store google-services.json file in project directory.
    - Update project level build.gradle to include following in the dependencies block:
      `classpath 'com.google.gms:google-services:3.0.0'`
    - And following line at the bottom of module-level build.gradle:
      `apply plugin: 'com.google.gms.google-services'`
    - Add dependencies for Admob and Analytics:
      `compile 'com.google.firebase:firebase-core:9.2.1'`
      `compile 'com.google.firebase:firebase-ads:9.2.1'`
- api is available at "https://www.udacity.com/public-api/v0/courses". It does not require any api_key or authentication.

Task 2: Implement UI for Each Activity and Fragment

- Build the portrait/mobile phone main activity UI as per screen 1 above. RecyclerView will be used to display the list items. The Main Activity will also have two Menuitems - one to refresh the conent and another to view favorite courses.
- Build UI for Detail view as per screen 2. It Will show additional details of the course. The user can scroll the view to see additional details. The user will be able to mark a course as favorite, watch course preview videos, if any, and view the course page on udacity website in a browser. The user will also be able to create a reminder (using menu action) in his calendar for the course being viewed in the details page.
- Build UI for Tablet layout as per screen 3. This will be created by reusing the fragments which were created for screen 1 and screen 2. The Detail layout may be tweaked to a bit to make the data presentation look better for this layout.

Task 3: Hookup Main Activity

This step involves creation of content provider and hooking up recyclerView adapter to fetch the data for display.

List of subtasks:

- Create the Main Activity layout without banner ads
- Create Content provider to store the data fetched through udacity api. Loading of fresh catalog data should take care to leave favorite courses untouched so that favorite course list is not overwritten by a refresh.
- Implement Test cases to testout the Content Provider
- Hookup periodic refresh of course catalog using "`Service`"
- Hookup recyclerView Adapter
- Implement search functionality
- Implement refresh menu action
- Implement "Show Favorites" menu action
- Handle the case when course catalog is empty and no network is available.
- Also Handle the case when no network connection is available and "refresh" action is initiated.

Task 4: Hookup build variants

This step involves completing the setup for build variants. The product will have two build flavours - one free with banner ads displayed at the bottom of each screen and another flavour - paid version which will not have ads.

List of subtasks:

- Create the alternate Main Activity layout with banner ads
- Setup project to use firebase banner ads as detailed above.
- Setup the build variants and create custom Android manifest and gradle build file to account for the fact that banner ads will not be shown in paid flavor

Task 5: Hookup Detail Activity

This step involves hooking up of Detail Activity and setting up all the actions.

List of subtasks:

- Create layout for detail activity as per screen 2. Create both the flavours - for paid and free versions.
- User Intents to get the id of Course details that needs to be shown and hook up with Content provider to get all the data.
- Hookup favorites action
- Hookup “Open in Browser” by starting an intent with action type of ACTION_VIEW
- Hookup “Watch Video” intent

Task 6: Implement Create Reminder Menu Action

This step involves enabling the “Create Reminder” functionality

- Create Menu item in detail activity menu.
- Create layout for “Reminder” dialog. It will have predefined text based on the Course Title and a date/time picker.
- Use Calendar Intent to push the details from dialog to the user’s calendar app.

Task 7: Implement Google Analytics

This step involves enabling google Analytics.

- Setup the Project to use Google Analytics.
- Setup functionality to track the courses for which the details are viewed
- Setup functionality to track the courses that are marked as favorites

Task 8: Implement Tablet Layout

This step involves setting up the Tablet Layout.

- Setup master/detail layout for tablets as per screen 3 above.
- Hookup the refresh of detail fragment on click on a new list item in the course list.

Task 9: Review Accessibility Features

This step involves cleaning up of accessibility features.

- Review the checklist at https://developer.android.com/guide/topics/ui/accessibility/checklist.html and ensure that accessibility guidelines are adhered to including description for all the buttons and images.

Task 10: Build Widget

This step involves building a widget for homepage to display favorite courses

- Build a widget to show data for courses that are marked as favorites by the user.
- Create a screenshot and update the widget preview image file.

Task 11: Implement Shared Elements Transitions

This step involves improving the user experience by implementing shared element transitions and other visual effects as per material guidelines.

- Implement selected item in Tablet layout.
- Implement Ripple on selection of list items - both in phone and tablet layouts.
- Implement shared elements transitions between main and detail activity elements.

Task 12: Get app production ready

This step involves building release builds of both the flavors.

- Generate apk signing key
- Review and turn off logging and debugging that is not required.
- Do a build for release-build apk file signed by private key generated above.