Description

Intended User

Features

User Interface Mocks

Screen 1 - MainActivity

Screen 2 – PlaySongActivity

Key Considerations

How will your app handle data persistence?

Describe any edge or corner cases in the UX.

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

Describe how you will implement Google Play Services or other external services.

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: Create AsyncTask and retrieve all necessary data about media files inside it. Then create RecyclerView adapters for each Activity/Fragment which uses this component

Task 4: Create Content Provider and SQLite database

Task 5: Create a Service which plays audio files and add notification

Task 6: Use ExoPlayer to provide video playback

Task 7: Create a PreferenceFragment for SettingsActivity

Task 8: Add google AdMob into the project

Task 9: Add Google Analytics

Task 10: Handle Error Cases, debug, test, optimize app

GitHub Username: DenisDavidek

UVPlayer

Description

UVPlayer groups most of your favorite features into one useful app. There is no need to have separate app for music or video playback. Play your favorite music or watch all your videos in one simple yet beautifully looking application.

Intended User

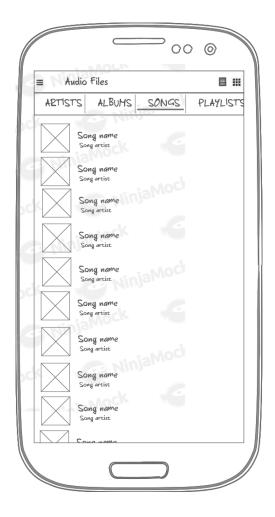
UVPlayer is an app for users who love playing their favorite media content from one place. App is for everybody who loves playing music or watching their favorite movies or shows from one place.

Features

- Scan for your media files (music and video)
- Music playback
- Add your songs to favorite playlist
- Interactive widget
- Notification for currently played song
- Video playback
- Changing the color theme of an app
- Being able to display media data in either list or grid layout

User Interface Mocks

Screen 1 - MainActivity



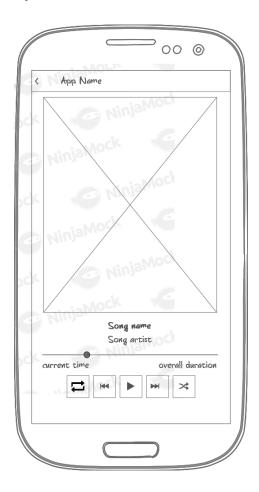
Main Activity screen – This screen is loaded when the app launches. It consits of NavigationDrawer which is used for navigation between each app sections.

There are also tabs which are used as a filter to filter between artists, albums, songs and playlist.

Below tabs there is a recyclerView which loads all necessary data into a list or grid of media items.

Two actions on right side of the actionbar is for changing the layoutManager of recyclerView.(list or grid)

Screen 2 - PlaySongActivity



Activity which plays currently chosen songs. There is toolbar with back button which after clicking will return user to the previously selected screen.

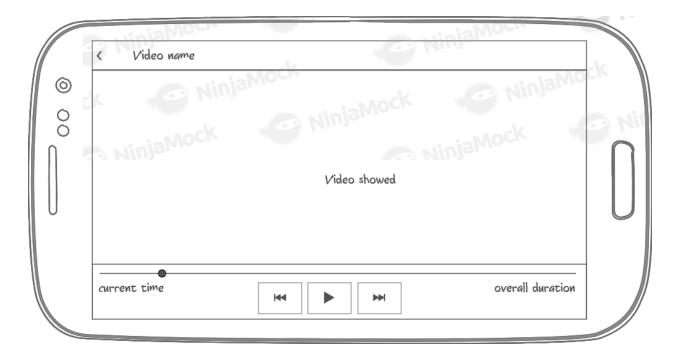
Below toolbar there is an album art of the currently played song.

Below album art there is song name and song artist textViews.

Below these textViews there is a progressBar for moving the playback of the song.

Below progressBar there is five standard buttons in media players: repeat, backwards, play/pause, forwards, shuffle.

Screen 3 - PlayVideoActivity



Activity which plays currently chosen video. There is a toolbar with back button which after clicking on it will return user to the previously selected screen. Inside Toolbar there is also a name of currently played video file.

Below toolbar there is a surfaceView where video is being played.

There is a bar layout on the bottom of the screen. It consits of progressBar, textViews for current time and overall duration and three buttons: previous, play/pause, next.

Screen 4 - SettingsActivity



Settings Activity – User can set if he wants to display media items in list or grid. They can also choose the overall color theme of the app from provided themes.

Screen 5 – Notification and Widget



There is a mockup for apps notification on the top of screen 5. It consists of album art, song name and artist and three standard buttons: previous, play/pause, next. Below notification mockup there is a mockup for widgets. It consists of all components used in notification mockup.

Key Considerations

How will your app handle data persistence?

I do get necessary data about audio and video files from android system content provider through MediaStore class. I will use this data for displaying in proper sections and in playlist section.

I will build my own Content Provider for the playlist section and use my own SQLite database for remembering which songs user has added to his favorite songs list.

I will use SharedPreferences for remembering if user prefers either grid or list layout option and I will remember color theme which user have chosen.

Describe any edge or corner cases in the UX.

Navigation through entire app is provided through NavigationDrawer component. There are a few categories inside: audio, video, settings and about app.

When user starts the application, application loads and displays found songs in categories – albums, artists, songs.

When user clicks on either artist or album then all related songs are displayed. When user clicks on a song new activity is launched which plays the song. Users can simply click on favorite button to add currently played songs into their favorite playlist. If user taps the back button, he will be returned into the last screen from where a specific song was chosen.

When user chooses video, application displays found videos.

When user clicks on a video new activity is launched which plays the video. If user taps the back button, he will be returned into the last screen from where a specific video was chosen.

Capstone_Stage1

When user chooses settings, they can set if they want to display media items in either list or grid layout. They can also choose the color theme for the whole app.

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

I will use ExoPlayer for providing video playback.

I will use Glide for loading images from local video files.

Describe how you will implement Google Play Services or other external services.

I will use Admob and implement banner ads inside MainActivity and activity which plays songs.

I will use Analytics to monitor how many users uses my app.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a project in Android Studio.
- Configure required libraries.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for ArtistsFragment
- Build UI for ArtistsDetailFragment
- Build UI for AlbumFragment
- Build UI for AlbumDetailFragment
- Build UI for SongsFragment

- Build UI for PlaylistFragment
- Build UI for PlaySongActivity
- Build UI for VideoFragment
- Build UI for VideoActivity
- Build UI for SettingsActivity

Task 3: Create AsyncTask to retrieve all necessary data about media files inside it. Then create RecyclerView adapters for each Activity/Fragment which uses this component

Create methods which returns all necessary data about audio and video files using MediaStore class. Use these methods inside AsyncTask to get all necessary data about mediaFiles and then after successful loading send it to RecyclerView Adapters.

Task 4: Create Content Provider and SQLite database

Create content provider and its database for favorite playlist. Create methods for inserting and deleting to content providers database.

Use loaders to load data from playlistContentProvider (to get the data) and then send it to RecyclerView Adapters mentioned in Task 3.

Task 5: Create a Service which plays audio files and add notification

Create a background/foreground service which provides audio playback and create notification.

Task 6: Use ExoPlayer to provide video playback

Go through ExoPlayer documentation and provide video playback into VideoFragment.

Task 7: Create a PreferenceFragment for SettingsActivity

Implement PreferenceFragment and all required functionality to remember which color theme user has chosen and if he prefers grid or list media item layouts.

Task 8: Add google AdMob into the project

Implement Google AdMob banners Ad into all Activities.

Task 9: Add Google Analytics

Implement Google Analytics.

Task 10: Handle Error Cases, debug, test, optimize app