GitHub Username:	https://github.com/Teldot
Project GitHub:	https://github.com/Teldot/Capstone-Project

PodStone

Description

Listen the best podcast from podcast.de without interruptions, save your favorite programs and more.

Intended User

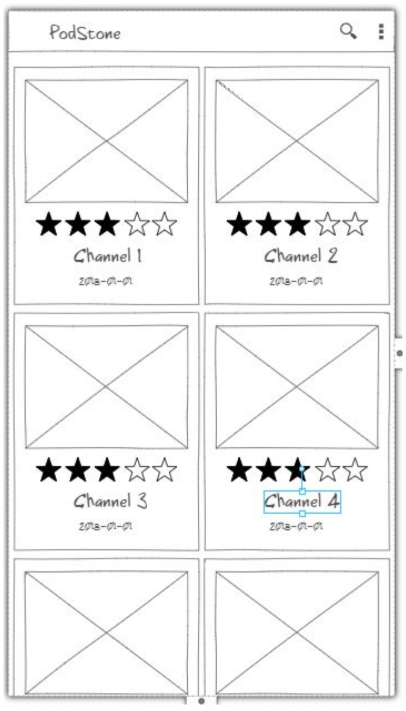
PodStone is for everyone who likes to hear podcast programs and wants an easy to use app.

Features

- Explore podcast.de, one of the biggest podcast directory.
- Filter by your preferred language (Languages available: en, de, es, fr, ru, ja, hi, el, pt).
- Save your favorites podcast.
- Easy access from home screen throw its widget.

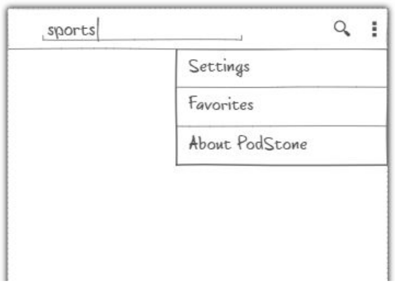
User Interface Mocks

Screen 1 - Main Screen



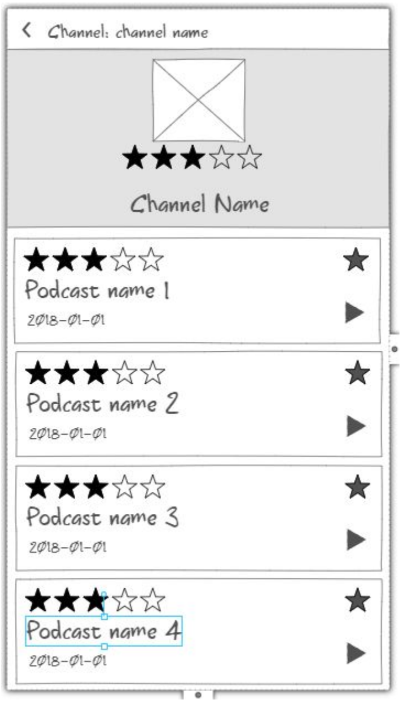
Main screen will show a list of new channels gotten from api.podcast.de/directory, displaying channel image (if it's available), channel rating, channel name and date. Additionally, it will have a SearchView Toolbar and a menu.

Screen 2 - Main Screen (SearchView toolbar)



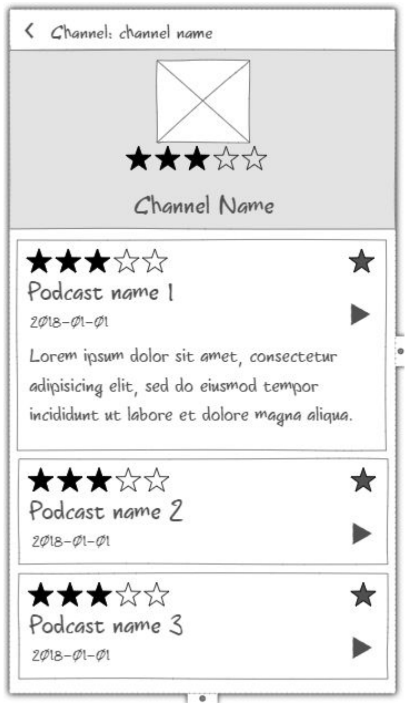
Main screen will provide a search option to allow users to find channels of their interest. Results will be showed in main screen recycled view. When search feature is canceled, list will load the initial directory. Either channel lists, directory and search will be filtered by language setted up on settings. The menu will show 3 options: ‘Settings’, ‘Favorites’ and ‘About PodStone’.

Screen 3 - Channel Screen



Channel screen will show channel image (if it’s available), channel ratting, channel name and the list of its podcasts with: podcast rating, podcast name, podcast date and two buttons to mark as favorite podcast and to play the podcast in an new activity.

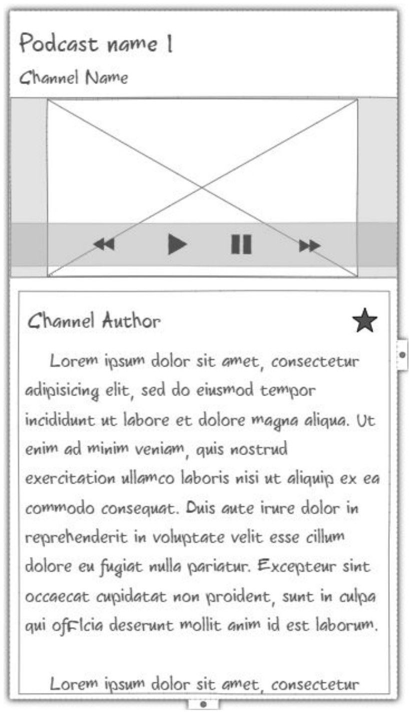
Screen 4 - Channel Screen (Podcast detail expanded)



Channel: channel name
Channel Name
Podcast name 1
2018-01-01
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Podcast name 2
2018-01-01
Podcast name 3
2018-01-01

When the user click on a podcasts list item, it will expand, showing the podcast detail.

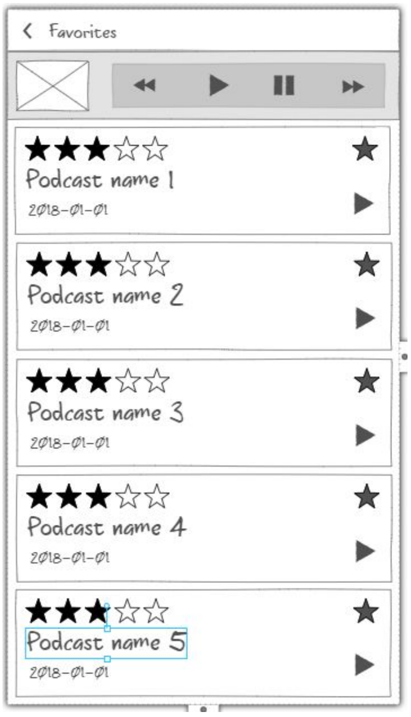
Screen 5 - Podcast Screen



Podcast name 1
Channel Name
Channel Author
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Lorem ipsum dolor sit amet, consectetur

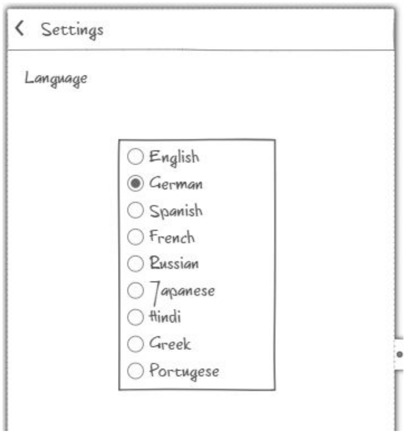
Podcast screen will show an ExoPlayer instance to play the podcast selected. This screen will show all related info about the podcast.

Screen 6 - Favorites Screen



Every podcast which user mark as favorites will be showed in this screen and it allows user to play them as a playlist.

Screen 7 - Preferences Screen

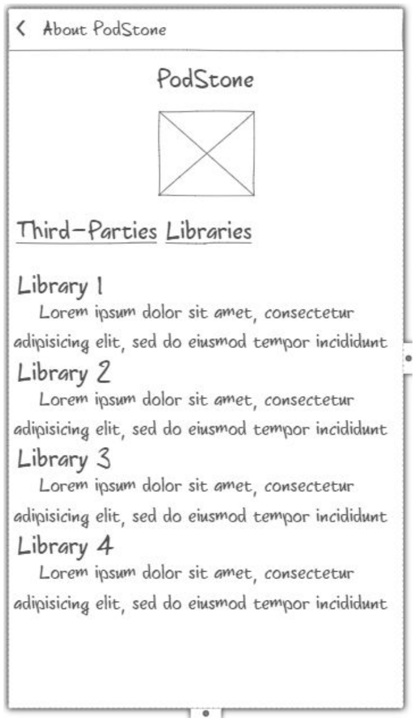


Language

English
[✓ German]
Spanish
French
Russian
Japanese
Hindi
Greek
Portugese

Preferences screen will allow the user to select the preferred podcasts language. Selected value will be used to request data to the API.

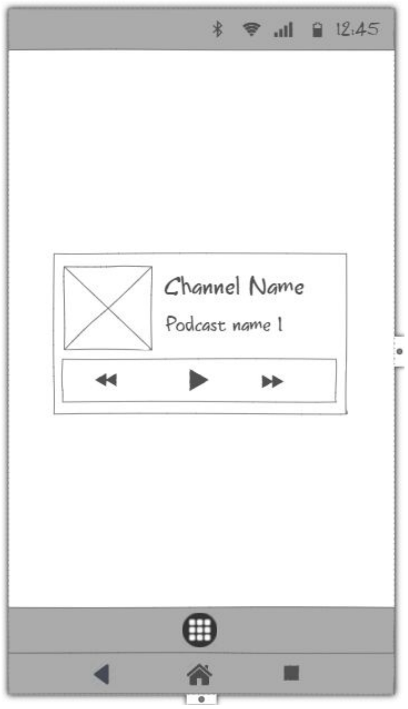
Screen 8 - About Screen



About PodStone
PodStone
Third-Parties Libraries
Library 1
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt
Library 2
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt
Library 3
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt
Library 4
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt

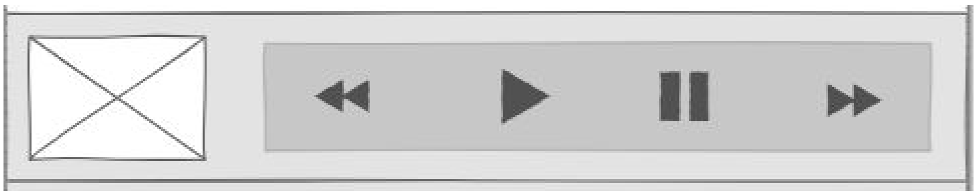
About screen will show related info about the app, including author, third-parties libraries, etc.

Screen 9 - Widget



Home Widget will allow the user to control media reproduction, showing podcast image, title and channel.

Screen 10 - Player View



Player view will be visible when PodStone will be playing podcast. It will show playing status, playing controls, podcast image (if available), and will allow user to go back to Podcast Screen when image is touched. .

Key Considerations

How will your app handle data persistence?

- Favorite podcasts metadata will be saved using a content provider in a local db.
- Preferred language will be saved in shared preferences.

Describe any edge or corner cases in the UX.

Once user has started playing any podcast, and goes back to keep searching more channels, a player view will be visible to show playing relevant info, and to let user go back to Now Playing View (Podcast View) when touches the ImageView on it.
When the player is started from a single podcast (in Podcast Screen), the player will back to Podcast Screen.
When the player is started from Favorites Screen, the player will play all the podcast in the list and it will return at this screen when the user clicks on the player view.

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

- Picasso will be used to load podcast images, because it allows you to load images easily with a few code lines.
- Exoplayer will be used as main media player because it is a powerful, customizable, and stable player.

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

- Admob will be used in this app implementing a banner Ad at the bottom of Main Screen, Channel Screen, Podcast Screen, and Favorites Screen.

- Analytics will be implemented in this app reporting app crashes.
- PodStone will get podcasts data from *podcast.de* service. API documentation available on https://api.podcast.de/.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create an Android Project called Podstone. Podstone project will be the main project.
- Add a module, an Android Library called playerservicelib. This library will handle the media player (Exoplayer) as a service and it will have a fragment (ExoplayerView) to connect to the service. This fragment will be used in the Main Screen, Channel Screen, Podcast Screen and Favorites Screen.
- In the main project, add all necessary dependencies, like:
    - com.squareup.picasso, com.google.android.exoplayer
    - com.google.api-client:google-api-client
    - com.google.android.gms:play-services-ads
    - com.android.support:recyclerview
    - and a reference to playerservicelib
- In the lib (playerservicelib) add necessary dependencies as well.
- Import the project into GitHub.

Task 2: Implement UI for Each Activity and Fragment

Main Screen:
- Load an initial list of most recent podcast available into a recycled view to show the user what can expect when they make a search.
- Include a menu with 3 options: ‘Settings’, ‘Favorites’ and ‘About PodStone’.
    - Setting will allow the user to set up preferred language.
    - Favorites will load pre-saved podcasts in a new activity.
    - About PodStone will show app’s information and third-party libraries.
- Include a search toolbar to allow the user to find podcast related to any theme. The results will be loaded into the recycled view in the main activity. The search results are loaded from podcast.de API/directory and
- Include a FrameLayout at the top of the Activity to contain the player fragment (implemented on later steps).
- Include a AdView at the bottom of the activity to show ads.

Channel Screen:
- We will need to show image of the selected channel, name, rate, and date.

- Additionally, this screen will show the list of episodes of the selected channel, so, we will need a recycled view.
- Each episodes will show its title, date, rate, and a button to select as favorite and a button to play it.
- At the top of this screen we will add a FrameLayout to contain the player fragment. This fragment will be shown only when it is playing a podcast.
- At the bottom of the screen we will include a AdView for the ads.
- When a channel is selected, it must expand to show its description.

Podcast Screen:

- This screen is shown when the user select a podcast and click on the play button on the Channel Screen. When it shows up, the podcast starts to play.
- Podcast Screen will show title, author, a player view, copyright, description and a button to add or remove from favorites.
- At the bottom of the screen we will include a AdView for the ads.
- The player will be the one in playerservicelib to use ExoPlayer as a service and keep listening to the audio when the user goes back.

Favorites Screen:

- This screen shows all the podcasts saved when user press favorite button on Channel Screen and Podcasts Screen.
- Podcasts list is shown into a recycled view.
- At the top of this screen we will add a FrameLayout to contain the player fragment.
- At the bottom of the screen we will include a AdView for the ads.

Preferences Screen:

- Preferences Screen will have a ListPreference to allow the users to select the language they want to make the podcasts searches.
- These languages are those allowed by podcast.de (English, German, Spanish, French, Russian, Japanese, Hindi, Greek, Portuguese).

About Screen:

- About Screen will show app info and third-parties libraries licenses.

Widget:

- App widget will show title and channel of podcast played.

Player View:

- Player view will use a custom player control (exo_playback_control_view.xml) to show previous, play, pause, and next buttons.
- Additionally, will show title and channel of the current podcast.

Task 3: Player Service Library

To allow the app to keep playing when the user leaves Podcast Screen we will need to implement the ExoPlayer into a service. This service will be bound by the player view into a fragment. Both components will be into the library to give to the app more independence.

- Create a class that extends Service and implements ExoPlayer. EventListener.
- Implement all required methods belonging to Service Class and ExoPlayer.EventListener.
- Add a method to add to list, remove from list, play media.
- Create an empty fragment that implements SimpleExoPlayerView.EventListener and an interface into the service class to handle service events.
- Add public methods to allow the container activities to control the fragment.

Task 4: Content Provider

We will need to implement a content provider to allow the app to save the favorite podcasts.

- Add a contract class with one table on it (ShowEntry), with its respective columns.
- Add a class as db helper to allow the app to create and update the database structure.
- Add the content provider class with its methods (query, insert, update, etc).
- We will need to query:
  - All the shows in database.
  - A single show info.
  - If an specific podcast is in the database.

Task 5: Asynchronous fetching task

Since we need to get data from internet, we need to use an asynchronous task.

- Create a class extending from AsyncTask and implements all required methods.
- Add the actions for:
  - Load a list at the app startup.
  - Load the search list.
  - Load the channel info.
  - Load the podcast info.
- Add methods to parse from json data, to app entities.

IMPORTANT: PLEASE MAKE SURE THE APP CONFORMS TO THE QUALITY GUIDELINES MENTIONED HERE

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”