
GitHub Username: pascalito007

My Podcats

Description

My podcast is a podcast player and manager that provides access to millions of free podcasts produced by podcasters. The app let you fetch feeds easily from iTunes or gPodder.net.
You can even download, stream or put your episode into your playlist and listen to them in your own way. You can subscribe to podcast and show episode or show podcast by category and even add podcast to favorites.
Users will receive notifications when new episodes are available to stream or download.

Intended User

My Podcasts app is intended for fans of listeners of podcasts or people looking to find new set of podcast to listen.

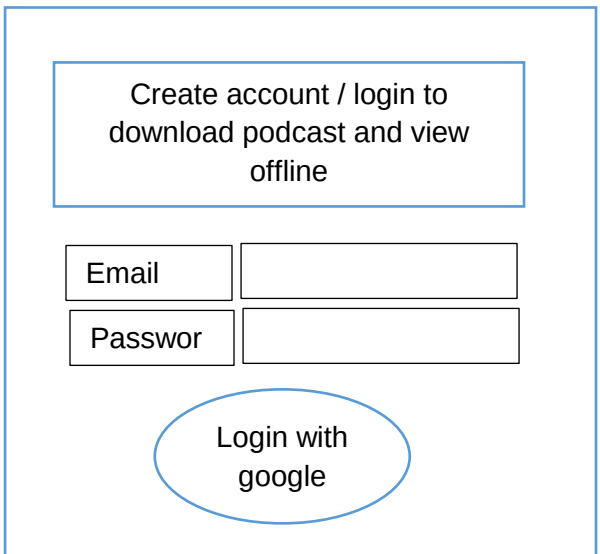
Features

List the main features of your app:

- Add and import podcasts from iTunes, gPodder.net
- Manage playback from anywhere: widget screen (optional), system notification and more.
- Add or remove podcast to favorite
- Download podcast and listen when offline
- Subscribe to podcast
- View podcast by category, episode
- View podcast description
- Offer free version with ads or paid version without ads
- Receive notifications of new episodes
- Create account or login to the app before downloading a podcast.
- Share podcast to friends

User Interface Mocks

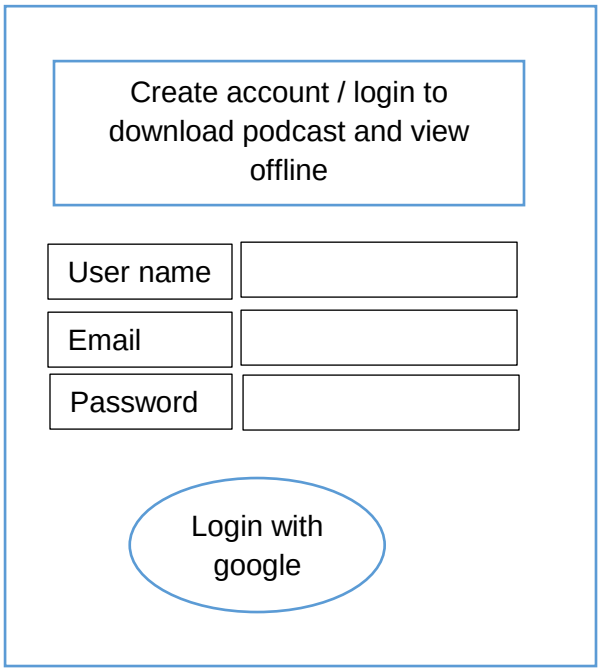
Screen 1 - Login



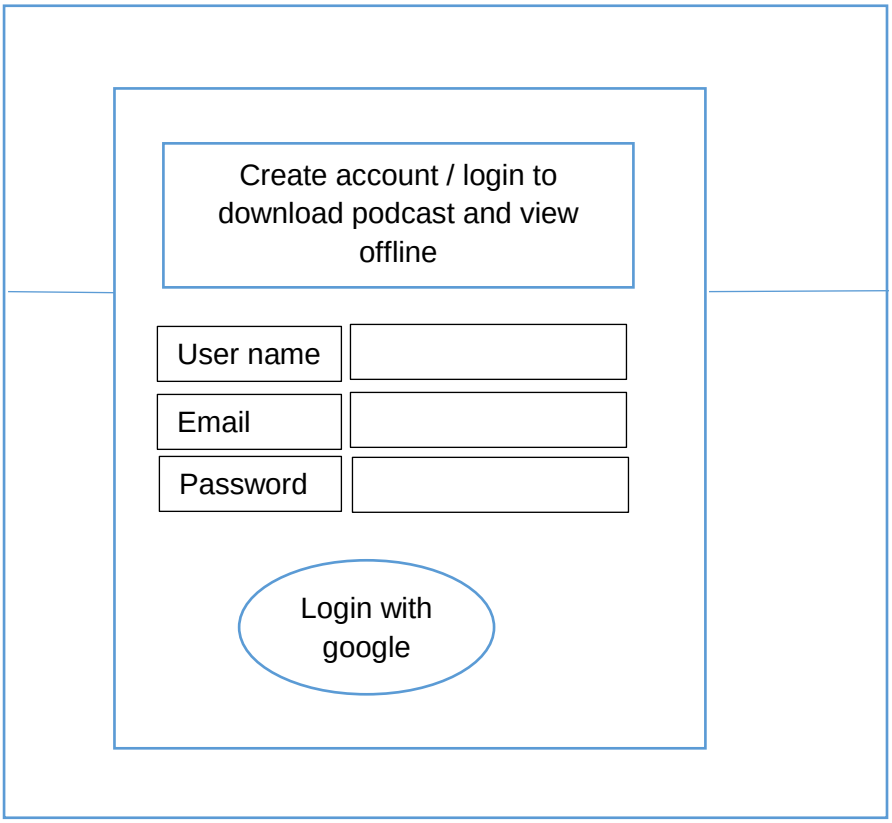
This screen is design to create login before downloading podcast

Screen 2 - Create user (phone and tablette)

Mobile phone

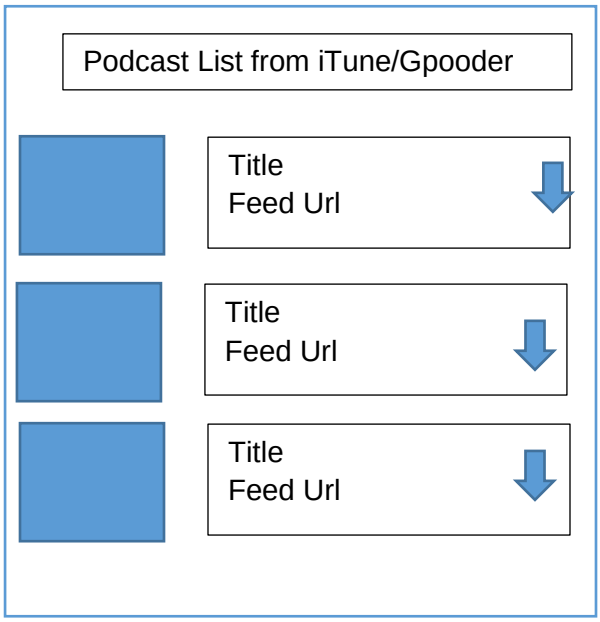


Tablette



This screen is design to create an account

Screen 2 - Podcast list

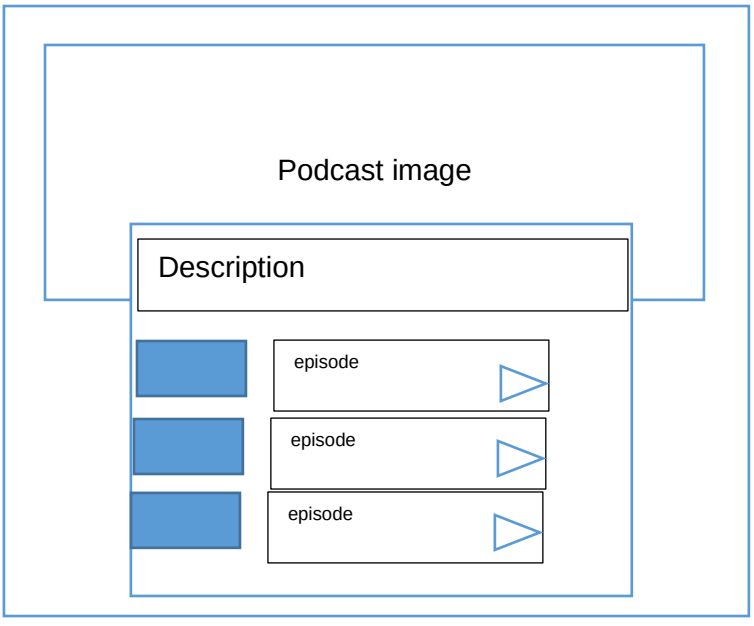


This screen is design to view all post fetch from iTune or Gpodder.net. On this screen user can download podcast after login

Screen 2 - Media player

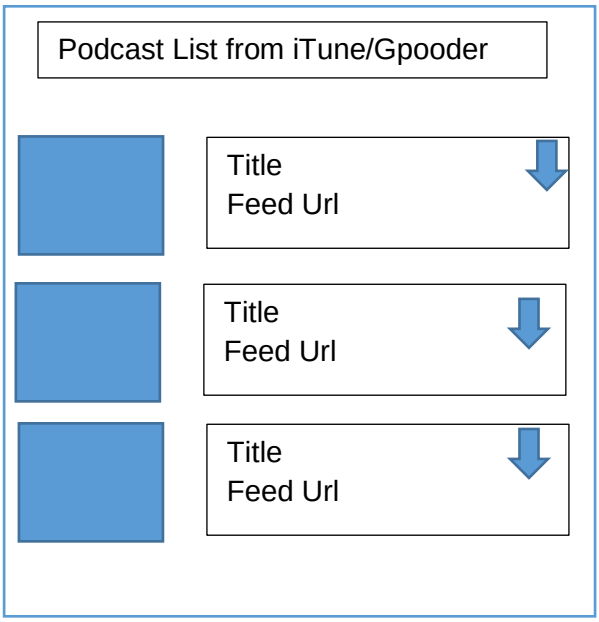
This screen is design to play podcast using media player and on top there is ads for free version of the app.

Screen 2 - Show details (phone and tablette)

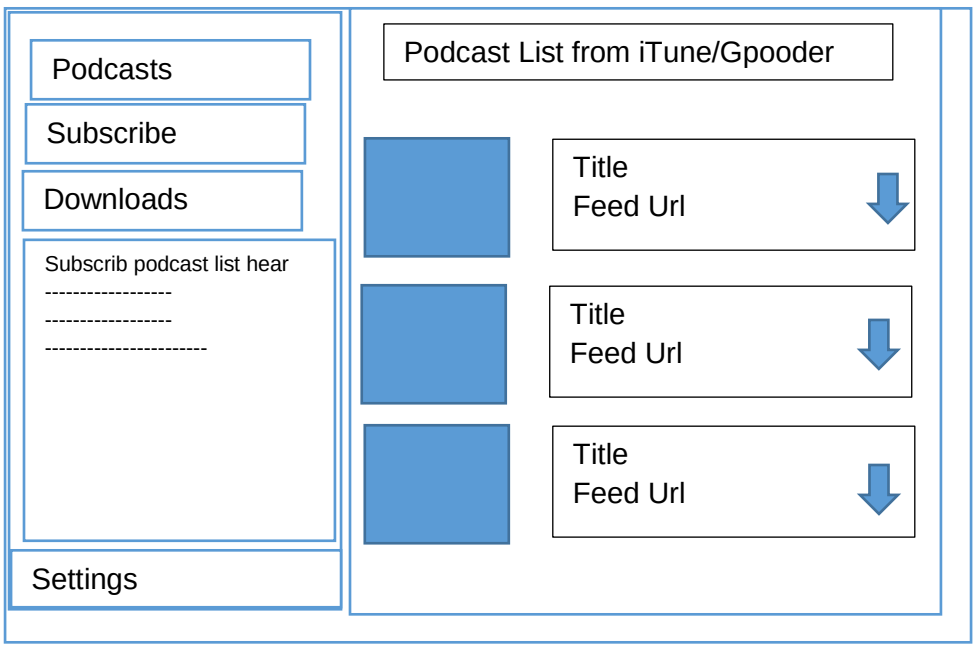


This screen is design to view podcast details.

Screen 2 - Podcast list



Tablette



This screen is design to view podcast list. On tablette options are shown in left of the screen.

Key Considerations

How will your app handle data persistence?

The app will use and AsyncTask to retrieve the rss feed for each provider (iTune, Gpodder.net) and save podcast subscribed data to a contentProvider, wich will store data in a sqlite database. The app will use Firebase to configure user creation and use preferences in the app to store credentials. Every time the app is launched, The cursorloader will fetch data from local sqlite database so that users can manipulate when offline.

Describe any edge or corner cases in the UX.

Since the MediaPlayer will persist across all screens through the Player in a BottomSheet, the user can return to the current episode information by expanding the Player.

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

I will use:

- Glide to handle the loading and caching of images.
- Firebase for user creation and users login before download a podcast. (I choose to let users create account before downloading podcast to give myself challenge on how to implement authentication in android app using Firebase).
- Espresso: to make some UI Test.
- Retrofit: to get data from and external api
- ButterKnife: To avoid using every time findViewById
- Schematic: To generate quickly content providers
- Facebook stetho: To debug the database

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

I am going to use Firebase and Admobe. Firebase to create new user account and achieve login and Admobe to show banner ads. I will use AdView from Admob to show the banner ad and google provider or email/password to create user using Firebase.

Next Steps: Required Tasks

Task 1: Project Setup

- Set up libraries
- Structure the project in packages: view, data, model, utils, adapter. Separate pure Java classes in separate pages.

.

Task 2: Implement UI for Each Activity and Fragment

Implies:

- Build UI for the main screen
- Build UI for the category fragment (subscribe, favorits,downloads)
- Build UI for preferences activity
- Build UI for podcast details screen
- Build UI for Adding podcast
- Build UI for Media player
- Build UI for user creation/Authentication Screen

Task 3: Design Local SQL Database

- Creating an SQL Database
- Creating tables holding episode, podcast.

Task 4: Produce Content Provider

Code logic for:

- Persisting episode and setting flags (downloaded, favorits)
- Retrieving episode, podcast, download podcast

Task 5: Implement Cursor Loader

The code behind each screen displaying iterative data retrieved from database must implement a cursor loader with a recycler view.

Task 6: Networking

I will use an Asynctask to fetch data from the iTune and Gpodder. And Intent Service to download podcast for offline use.

Task 6: Implement Admob

Add dependencies and show adView in all free activities.