GitHub Username: mladenbabic

CoffeeRoom

Description

Love to drink a coffee sitting on a porch or in your favorite room and reading your local news? Then this CoffeeRoom news app is for you. With new material design and lot of features like animations and with great user experience, you can access and get at glance favorite news. Stay in the know with breaking headline news, the latest entertainment stories, exclusive lifestyle features, and your local news wherever you go.

Intended User

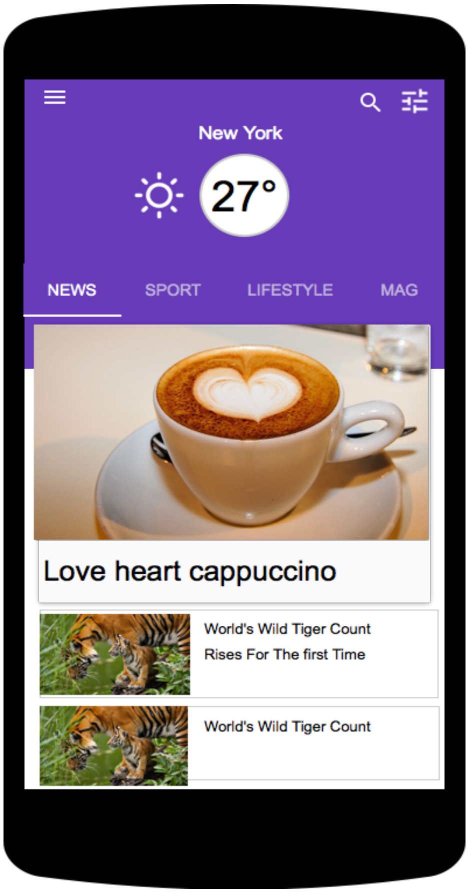
This app is for everybody who likes the latest material design look and feel and who likes to be in always in the know.

Features

- Material design look
- Customized news layout
- Weather functionality
- Search functionality
- Picture gallery
- Youtube Video
- Share article functionality
- View and add comment on an article
- News widgets
- Push notifications for breaking news (Simple and Expandable)
- AdMob
- Google Analytics
- Share and social media

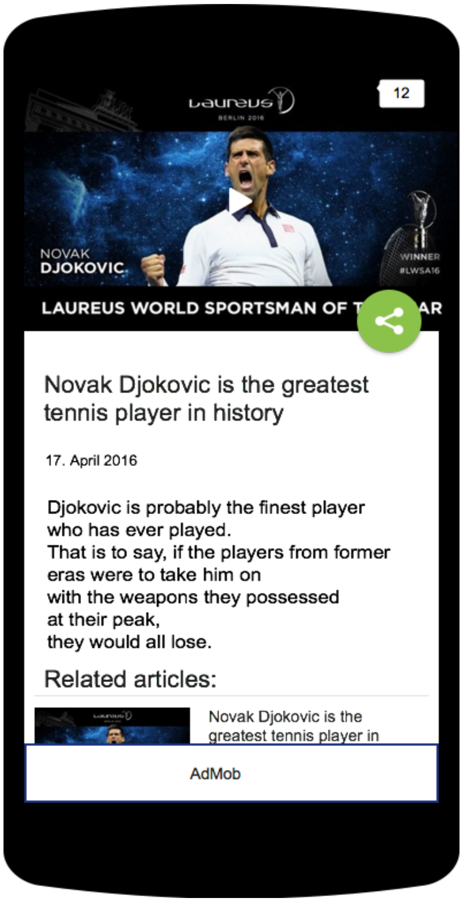
User Interface Mocks

Home screen 1



Home screen with the list of first 10 newest articles (Endless scrolling)

Article Details



Novak Djokovic is the greatest tennis player in history

17. April 2016

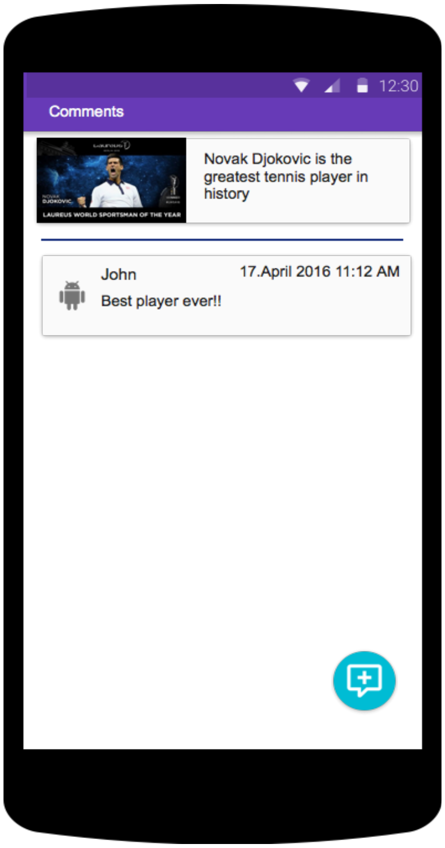
Djokovic is probably the finest player
who has ever played.
That is to say, if the players from former
eras were to take him on
with the weapons they possessed
at their peak,
they would all lose.

Related articles:

Novak Djokovic is the
greatest tennis player in

Article detail screen with article details and related articles based on same category.

Comments



Novak Djokovic is the greatest tennis player in history

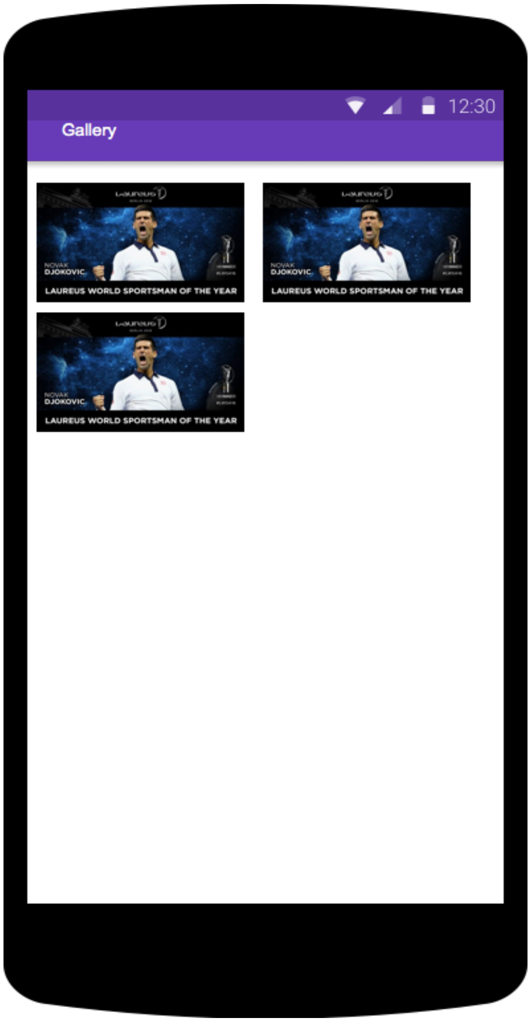
John

17.April 2016 11:12 AM

Best player ever!!

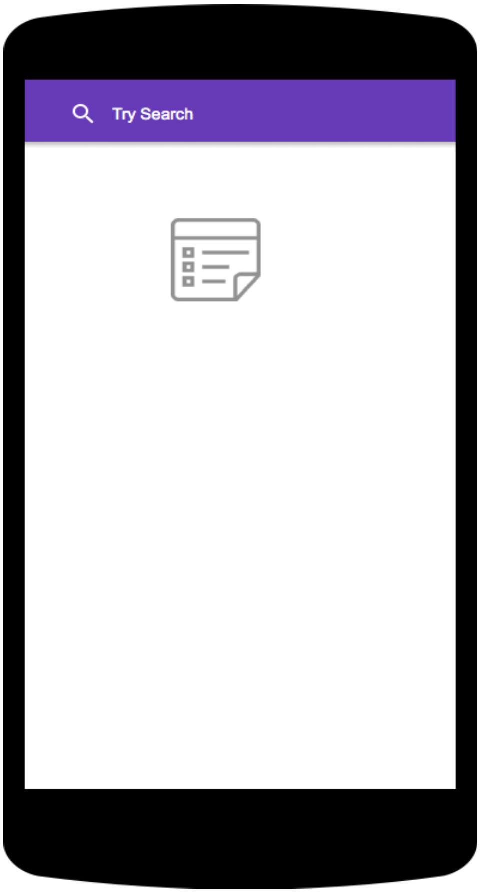
Comments screen with the list of comments and floating button for adding a new comment

Gallery screen


12:30
Gallery
NOVAK DJOKOVIC
LAUREUS WORLD SPORTSMAN OF THE YEAR
NOVAK DJOKOVIC
LAUREUS WORLD SPORTSMAN OF THE YEAR
NOVAK DJOKOVIC
LAUREUS WORLD SPORTSMAN OF THE YEAR

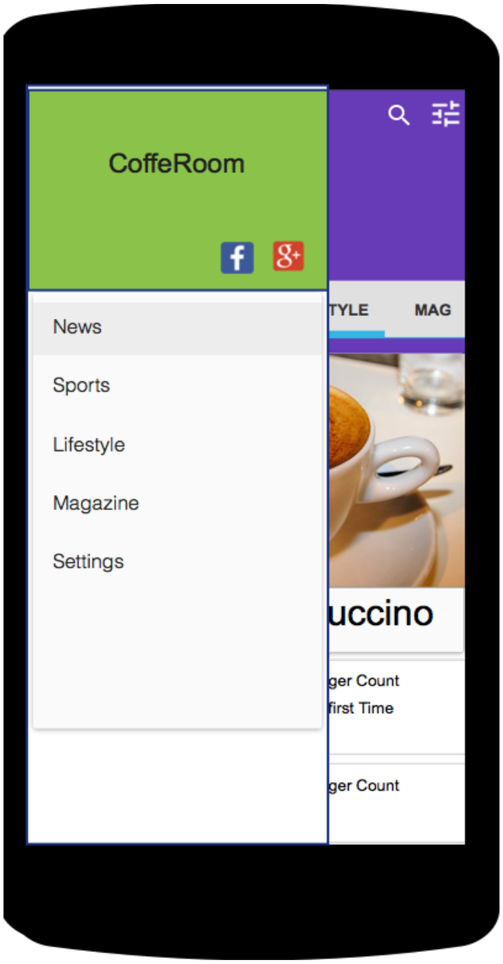
List of images related to the current article

Search screen



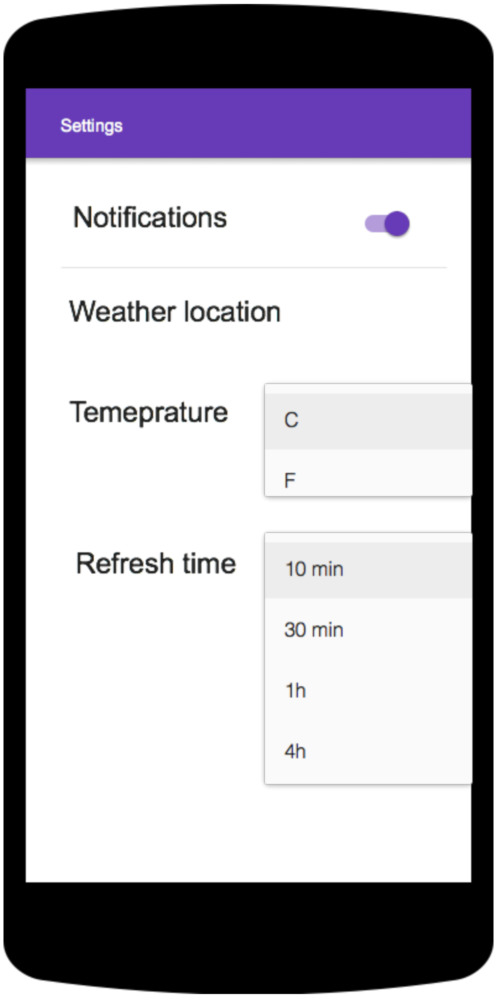
Screen for searching articles based on the text

Navigation drawer on home screen



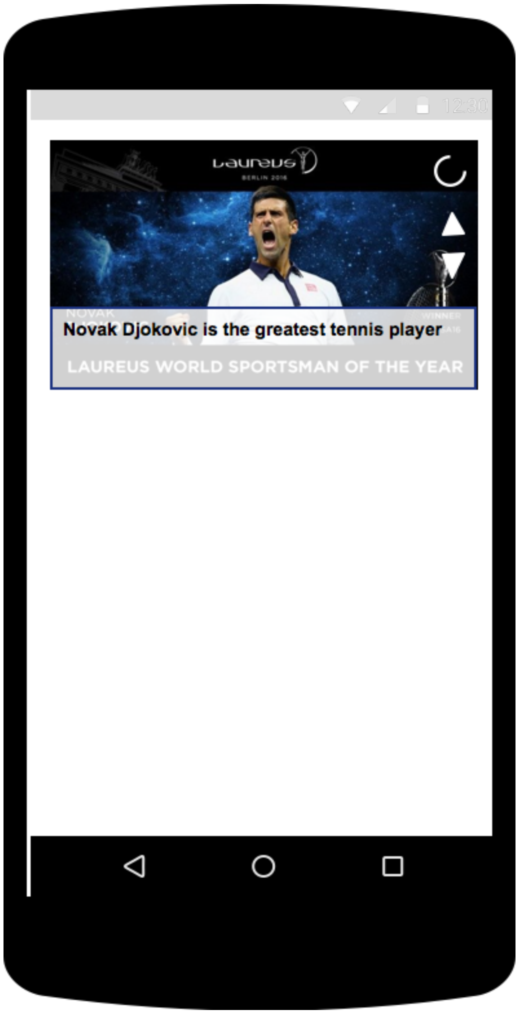
This screen shows navigation drawer on the home screen

Settings Screen



Screen shows settings options

Widget on Home Screen



Novak Djokovic is the greatest tennis player

Key Considerations

How will your app handle data persistence?

All data will be downloaded from the web server. No need for saving data in the locale database.

Describe any corner cases in the UX.

On the home screen, the user is able to infinite scroll articles in the list. By selecting an image or article title the user will be navigated to the article detail screen with the transition animation.
Tap on weather temperature will update weather for selected location or current location which depends on the user preferences.
The article detail screen contains the article content and related articles with the same category as selected article.

A user is able to navigate to a list of comments and/or add a comment for the current article.
List of added pictures for current article is also available.
Back button from the article detail screen will back the user to selected position on the home screen.

A user could search articles by clicking the “magnify” icon on the action bar. The search action leads the user to the search screen with article results. Back button navigates the user back to the home screen.

On the settings screen, the user disables/enables a push notification, sets weather location, sets the temperature unit and an interval for a weather update.

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

Gson - converts from json to pojo object and vice versa
Butterknife - field and method binding for Android views
Calligraphy - Custom fonts for Android
Retrofit - Handles Http requests
OKHttp - Http client for Android and Java apps
Picasso - Handling the loading and caching of images.
EventBus - Simplifies communication between fragments, activities and services.
Nineoldandroids - Animation libraries for old Android version
Android reactive location - Small library that wraps Google Play Service API in RxJava
RxJava - Library for composing asynchronous and event-based programs using observable sequences for the Java VM
RxAndroid - This module adds the minimum classes to RxJava that make writing reactive components in Android applications easy
MaterialViewPager - Material designed a ViewPager
Convert-Gson for Retrofit - Gson converter

Task 1: Project Setup

The first steps will be to set up libraries. Then, next step will be to create Util classes for handling HTTP web requests with a web server. The next steps will be:

- Create model objects representing data retrieved from the web server
- Create class with methods for retrieving data from the web server
- Create Web Services API for data
- Create Yahoo API Key for retrieving weather data

Task 2: Implement UI for Each Activity and Fragment

- Create Splash screen
- Create MainActivity and fetch data from server
- Build UI for MainActivity.
- Create Fragment for MainActivity and layout for list of articles
- Create list item layout for the list of articles
- Create DetailActivity and UI for it.
- Create SearchableActivity and layouts for search result.
- Create SearchArticle Fragment for search result
- Create SearchLocationActivity for retrieving places data using Google Place API.
- Create a layout for search places.
- Create SettingsActivity and retrieving data from SharedPreferences
- Create SettingFragment for showing settings data
- Create UI for SettingsActivity
- Create CommentsActivity
- Create CommentFragment and build UI for the comments list
- Create layout for list item for the comments list
- Create ImageGalleryActivity for showing article image and UI layout
- Create Widget and layout for it.
- Create simple and expanded notification layout

Task 3: Implement Google Play Services and Permissions approvement for the location

In this step enable and used Google Play services and handling run time permissions

- Enable Places API in Developer's Console

- Enable GCM API in Developer’s Console
- Enable AdMob API in Developer’s Console
- Enable Google Analytics in Developer’s Console
- Handling permission dialog for the location

Task 4: Logic implementation

- Implement push notification functionality using GCM API
- Implement AdMob functionality
- Implement Google Analytics
- Implement a material view pager
- Implement a navigation drawer
- Implement a list of articles based on category
- Implement an endless scrolling on home screen
- Implement navigation to the article detail view
- Implement a service for weather update
- Implement a weather view and weather tap functionality on home screen
- Implement a search article functionality
- Add Transition Animation between home screen and article details view
- Implement functionality for retrieving an article data from the server and show it in article detail view
- Implement a share functionality on the article detail view
- Implement a function for retrieving an image gallery in the image gallery view
- Implement a function for showing a video related to the selected article on YouTube
- Implement functionality for fetch comments data for the selected article
- Implement add comment functionality for selected article
- Implement a share for social media in the navigation drawer
- Implement add places functionality for the weather settings
- Implement notification option in the settings view
- Implement a weather settings options in the settings view
- Handle data obtained from GCM and add notification functionality
- Implement logic for retrieving places using Google Places API
- Handle error cases

Task 5: Implement a widget

- Implement a tap functionality for scrolling the newest articles in the widget
- Implement a tap functionality for manual updates of the newest articles in the widget