**How to Use this Template**

1. Make a copy [ File → Make a copy... ]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

**Submission Instructions**

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

---



**GitHub Username**: @setico

# AfricaRDV

## Description

Stay informed of the latest events or news in Togo and  all the Africa on your Android smartphone and tablet. Follow the live news and check news through our many thematic headings : Highlights , Society, Politics , Development , Economics, Peace and Security, Africa , Sport etc ...
Get the news real-time notifications and share content you like in a few clicks with your loved ones.
The design and navigation have been completely revised and optimized to offer a seamless experience through a streamlined interface .
Africa appointments , Africa on his appointment.

## Intended User

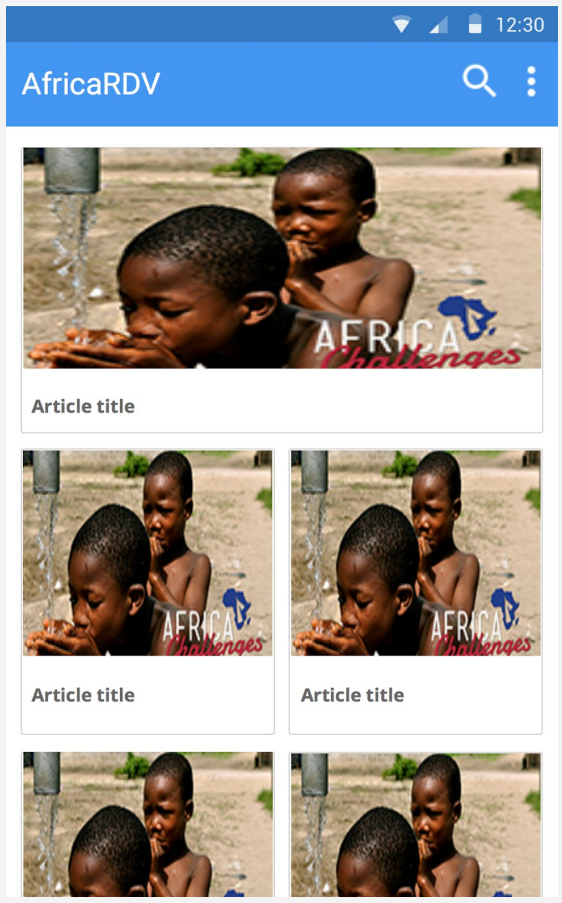
This application is intended to all persons who wish to receive daily news on his country or africa in general.
it will be very useful for diplomats, journalists, economists, politicians etc ...

## Features

- Sync the 50 recents posts on AfricarRDV website (www.africardv.com) via wordpress Json API on demand or automatically.
- Display a list of 50 recents post from AfricaRDV website on home screen.
- Save data locally to make user access and read article in offline mode.
- Using of syncAdapter to make periodic and automatic synchronization.
- Use of shareActionProvider to allow article sharing by users.
- Display article content to user with it cover picture, tags, category and author details.
- Make local articles searchable by users.
- Widget on Home, to show the last recent article.

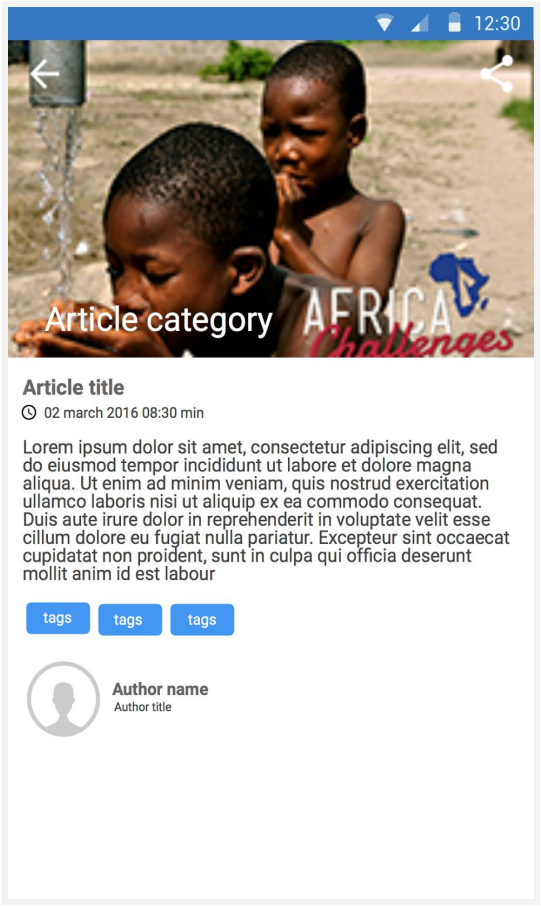
## User Interface Mocks

## Home Screen



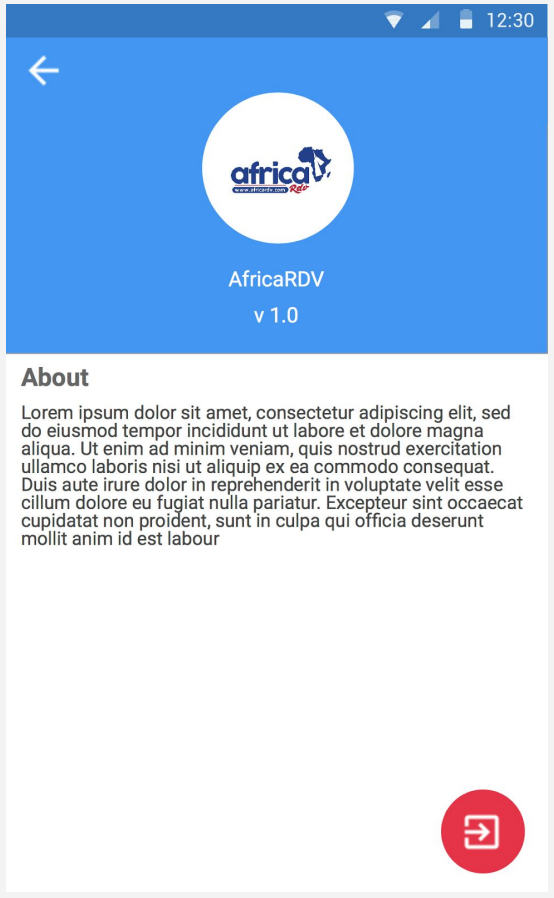

Using of reclyclerView and SwipeRefreshLayout to show articles list.

## Article details Screen



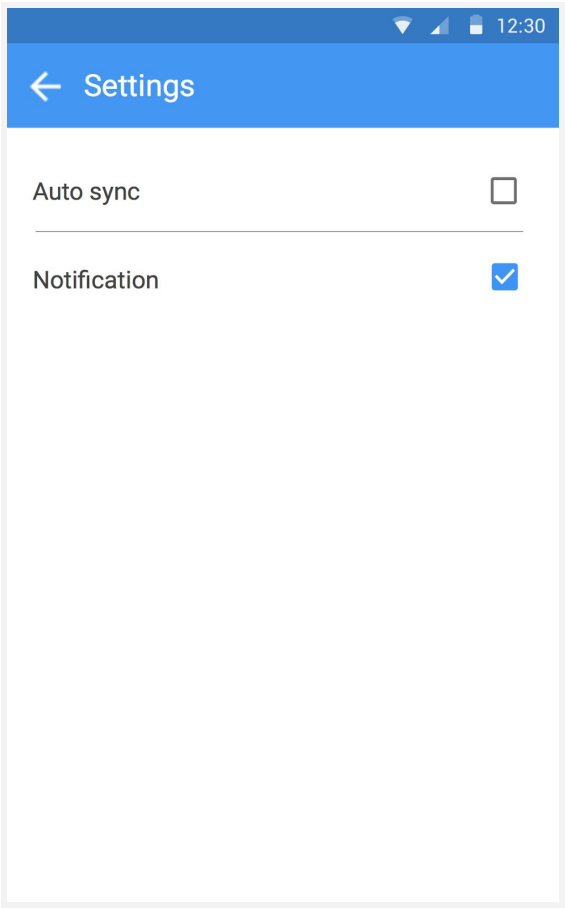

Here, Article details screen, using of material design stuff and making parallax effect on article cover, using of custom view to auto organize article tags and ShareActionProvider to share content with other applications.

## About Screen



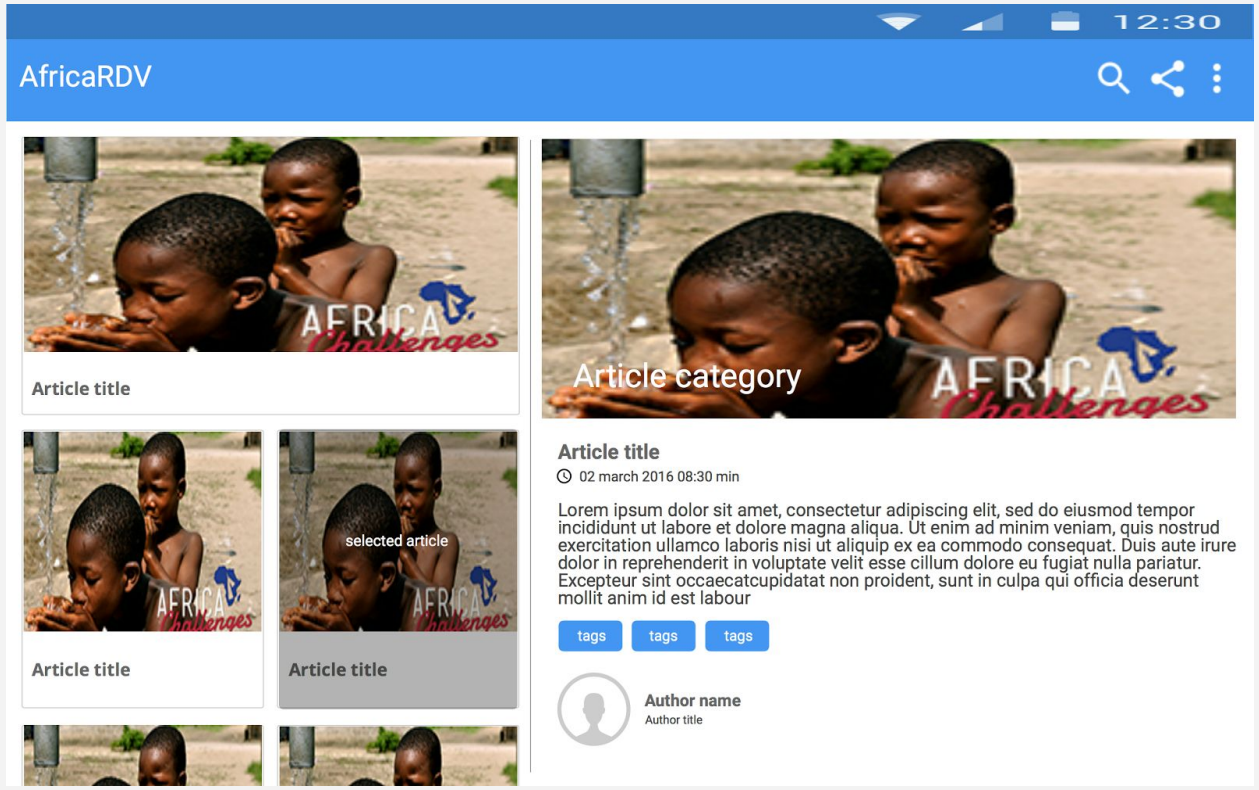

This screen is the about screen of our App. And here you have brief presentation of AfricaRDV, an FloatingActionButton to open directly in browser AfricaRDV website.

## Settings Screen




Settings screen, here you can allow app to show or not notification on update, activate or desactivate automatic sync in app.

### Home Screen Tablet mode (Landscape)




We show here the articles list in the first pane and selected article details in second pane. this screen is available only on tablet.

## Key Considerations

### How will your app handle data persistence?

In this App we create a custom Content provider (AfricaRDVProvider). When the app sync recents post from AfricaRDV website, we parse a Json Api content and then save it into Sqlite Database through our custom Content Provider.
When user pull manually the home screen from top to bottom, with start a PullToRefresh request and sync again data from server.

If auto sync is activated in setting, the app can make a sync automatically via sync adapter and user get notified when there is somethings new to save.

### Describe any corner cases in the UX.

- No internet connection – show message
- No data available to show on the Home screen – show message
- No result from search – show message
- Can't load article cover picture - show log

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

- Glide, to process article cover picture, I'll use this library. Because it is so powerful library to access images, handle and caching image for offline using.
- Google Support Libraries, to support Material Design and fluid UX
- Okhttp, to handle efficiently http request to server
- PugNotification, to build a nice and rich notification for android version up to 2.3
- Google Analytics Library - to get statistics informations about user's
- Google Admob Library – to show ads for monetization purpose.

## 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

- Configure libraries:
    - Google support librairies
    - Google Analytics(Google Play Services)
    - Google Admob(Google Play Services)
    - Glide library
    - Okhttp library
    - PugNotification library
- Add App permission to AndroidManifest.xml

    - INTERNET
    - ACCESS_NETWORK_STATE
    - READ_SYNC_SETTINGS
    - WRITE_SYNC_SETTINGS
    - AUTHENTICATE_ACCOUNT

- Add Play Services version meta data and also Admob interstitial Activity declaration into AndroidManifest.

## Task 2: Implement UI for Each Activity and Fragment

List the subtasks:

- Build UI for MainFragment used in Home Screen
- Build UI for MainActivity with master/details support for tablet
- Build UI for Article details fragment and activity
- Build UI for AboutActivity
- Build UI for the App setting activity
- Build UI for App Widget

## Task 3: Implement Activities and fragments

- Run App first time
    - Build and run initial project to check if everythings is ok.
    - Create database and custom Content Provider for saving data ( article, category, tag, author, attachment)
    - Create models
    - Create contract , content Provider, DbHelper class.
- Loading data from wordpress Json Api and save to SQliteDatabasecreate
    - An intent service with an implementation of AsyncTask to make network request.
    - Use okHttp for handling http request to api.
    - Create a custom Json parser to get data from json
    - Save data or update it via content provider
    - Implement PugNotification to notify user when new articles were available in app ( notify with article count if there are many and notify with article cover if only one article is available)

- Syncing data from Json Api automatically
    - Implement SyncAdapter with authenticator

- SplashActivity
    - Implement a splash Activity with AfricaRDV logo

- MainActivity & MainFragment
    - Implement a RecyclerView and it adapter
    - Implement PullToRefresh on activity
    - Load last 50 articles from database via content Provider into the recyclerView
    - Using of Loader to load data in background
    - Configure master/details support on MainActivity
- DetailActivity & DetailFragment
    - Load selected article and show it
    - Implement shareActionProvider for sharing article data with someone
    - Create custom View for auto organizing article tags
- SettingsActivity
    - Configure setting option for auto syncing, if checked the SyncAdapter will start Automatically for checking update on server.
    - Configure notification option, show notification to user only if checked.

## Task 4: Create Widget

- Implement a simple widget to show last article on the home screen.

## Task 5: Configure Google Analytics

- Implement Google Analytics for stastics
- Analytics by screen name
- For elementary information not for get all information

## Task 6: Monetization strategy

- Implementation of Google Admob

## Task 7: Make a full test before launching

- Handle UX error
- Settings
- Content

---

**Submission Instructions**

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"