How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]

2. Name your document file: "**Capstone_Stage1**"

3. Replace the text in green

GitHub Username: **felzahra**

Fatma News

Description

Fatma News is an Android app that allows users to read their favorite technical news headlines everyday and share it with their friends easily. Users can also save the news they like for offline reading.

Intended User

This app is intended for anyone who is a technology lover and wants to keep himself abreast with the latest technical happenings in the world.
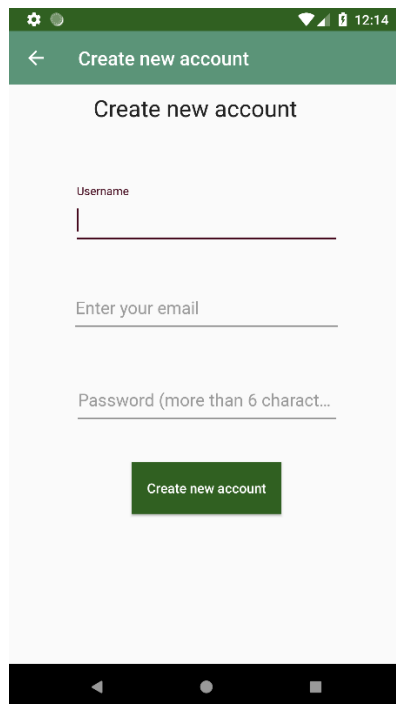
Features

List the main features of your app. For example:
- Read latest news updated everyday.
- Ability to choose different sources for technical news.
- Favorite/Bookmark news headlines that you like so that you can read them later.
- Sort news by top, latest and popular.
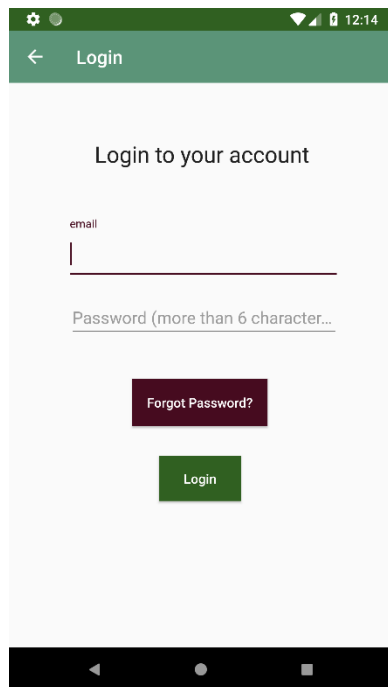
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.
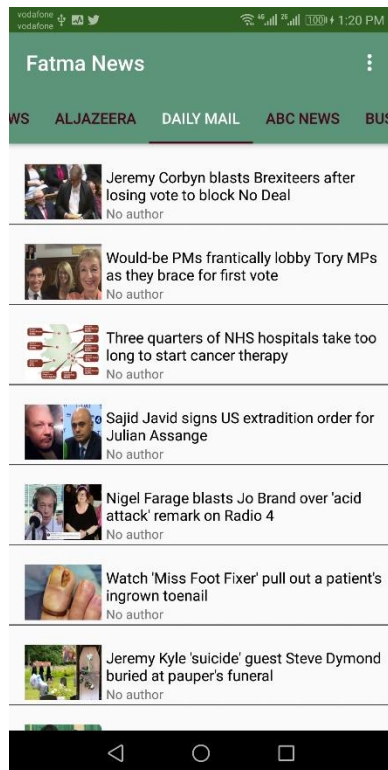
Screen 1

Screen 2

The main screen which shows a list of news headlines with each item shows the image of the news url.

Screen 3

It also has favorite button to allow users to favorite/bookmark the news. The share fab allows users to share the news with their friends.
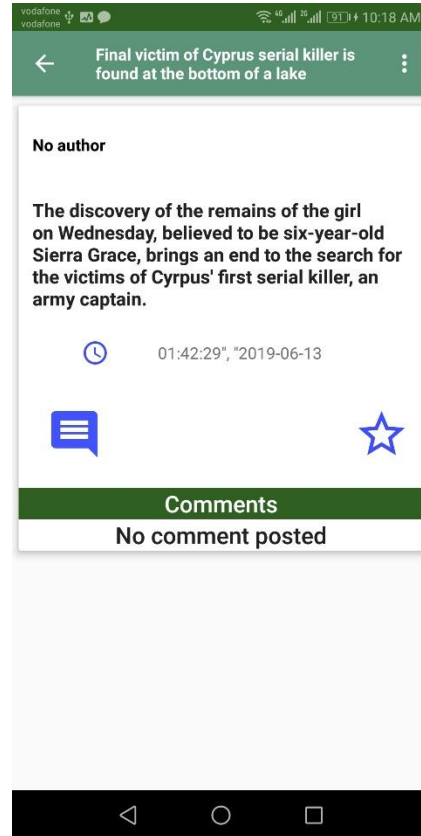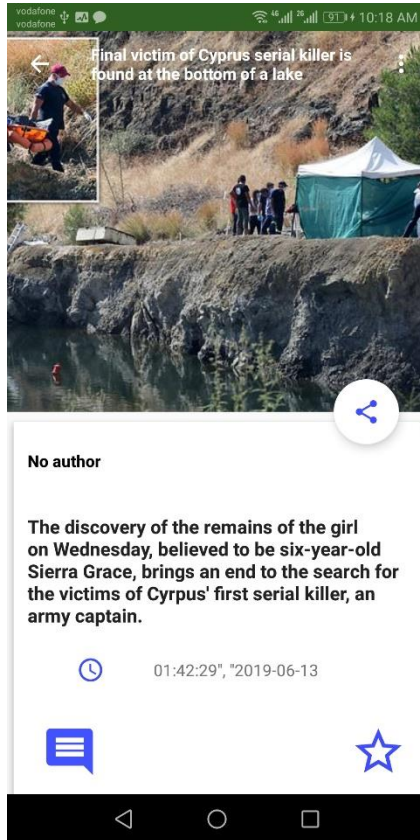
Key Considerations

How will your app handle data persistence?

Data persistence will be handled using Firebase.

Describe any edge or corner cases in the UX.

A corner case would be rotation of the device when the news is being loaded.

Incorrect passwords-emails in login : will be get a message that this data is not right

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

- Picasso to handle the loading and caching of images.
- Butterknife for view binding
- Firebase
- Espresso

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

Sign in with google : Login with email or password or even google gmail account
Google ads : adding ads to main screen
Firebase : Analytical , authentication , notification and realttime database

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

First I will initialize a new project in Android studio and do the following steps
- Add all the required dependencies
- Make proper packages for easy code maintenance.

Task 2: Implement UI for Each Activity and Fragment
- Build UI for login system (login-reegister-forget)
- Build UI for MainActivity
- Build UI for each fragment in MainActivity
- Build UI for DetailActivity
- Build UI for favorite Activity
- Build UI for add comment

Task 3: Implement the Main Activity

- Write the code when the user Forgot/login/register Password.
- Write the code for showing the journals on main screen
- Write the code for showing news articles in each journal activity
- using butterknife to bind the view
- Use Picasso to load news
- Show and populate the news data in a recycler view.
- Quering the database to check if the specific article has comments.

Task 4: Using loaders, handling Data persistence and Background Tasks

- Write code for Detail Activity.
- Implement Loader to load favorite news data from database.
- Use Firebase to regularly update data in the background
- Implement sharing feature when the user clicks on share fab.
- Handling the add/remove news part in favorite database when user presses the favorite button.

Task 5: Clean up and finalizing

● Clean up the code and move all strings to strings.xml
● Check for error cases, app crashes and do the required error handling

Add as many tasks as you need to complete your app.

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**"