

Coursework One SNews Design

Samuel Netherway

1 Introduction

Staying informed on the go is one of many smartphone functionalities which we all take for granted. Smartphone news applications help us get our daily dose of the on-goings of the world around us in a concise and user-friendly format. It is therefore clear to see how a well-designed news aggregator app can easily take the app store by storm. This document will detail the design phase of my app “SNews” (Smart News) and its place in the variety of news apps currently on the android market. I will pay special attention to user interaction, android material design principles and implementation strategies.

2 Core Functionality

Every news app must provide several core functions for a meaningful user experience. Therefore, I decided to investigate a variety of IOS and android news apps to identify key design decisions. Based on my observations I have decided that my app will contain five screens, home, discover, games, search and profile which can be navigated between with the use of a bottom navigation menu. My decision to position the navigation system at the bottom of the screen promotes ease of use with a single hand and removes the need for screen titles (see Figure 1). Each screen will follow android material design practises to provide a fluent experience which is described in further detail in the sections below.

2.1 News Feed

The home screen shown in Figure 1 demonstrates how multiple news articles are presented in three distinct layouts. The top portion of the screen will contain a horizontal scrolling view with several toggle buttons each representing a different topic or interest. Any combination of toggle buttons can be selected / deselected for full customisation of the news feed contents. Having a topic selection mechanism positioned at the top of the screen is a popular design decision by many news apps such as the Mirror [1] and The Sun [2]. I believe that its inclusion in my app will provide a sense of familiarity and will therefore reduce the time spent learning how to use the app. The main body of the screen will be occupied by a recycler view containing small, medium and large previews of articles which can be clicked on for further expansion. If an article is selected, the content will be presented in a web view or if in offline mode (see section 3.3) a custom news presenter with reduced dynamic content such as images, videos, etc.

2.2 Finding Articles

The discover screen will provide users with the opportunity to customise their news feed. Topics which interest the user can be added to the horizontal scroll view on the home screen to provide easy switching and customisation. The discover screen will contain a list of categorised subjects such as technology, science, sport, etc. Positioned parallel to each subject, a toggle button will control whether the subject appears in the horizontal scroll view discussed in section 2.1. Users will also be able to find news providers, topics, locations, articles, etc with a search bar which will list content in a basic recycler view.

2.3 Personalisation and Alerts

The inclusion of a profile section will act as a central hub for viewing account details and customising almost every adjustable aspect of the app. This will allow users to control the rate at which new articles are fetched, alert preferences, and other factors mentioned throughout the document. Alerts will inform the user when new spotlighted articles are available (see section 3.2) and when articles are downloaded and deleted (see section 3.3). Various attributes such as frequency, quiet times, and even scheduled alerts for each topic can all be adjusted in the user profile section of the app. I believe that it will provide the user with full control over all notifications which will reduce the potential of frustration and confusion.

3 Bonus Features

3.1 Bonus Feature One: Multiple Choice News Games

For my first bonus feature, I wanted to develop a less traditional method of getting informed. I have decided upon the inclusion of a games section within the app. This will contain a series of multiple-choice games such as, guess the article title from the image, guess the category of the article and guess the missing fact. At the end of each question, an opportunity to view the original article will be presented to the user if they decide to find out more before continuing to further questions. This will provide an interactive and entertaining method of presenting specifically tailored news topics to the user in an informal manner. The use of multiple-choice games with simplistic user interfaces, ensures that the effort required to use the app is still low, meaning that users will find the games enjoyable opposed to a chore. Additionally, options to cheat (view article) and skip will be available to reduce user frustration (see figure 2).

Prior to implementing the games section of the app, I researched the idea of generating multiple choice questions from article titles and descriptions to confirm the feasibility and scope. Based upon the information returned by a sample news API [3], I believe that I will be able to implement some games such as match the picture with the article title, but other games such as guess the missing fact may require the use of an API. If generating suitable multiple-choice questions and answers is beyond the scope of the project, I will make use of a suitable API such as Quillionz [4].

The inspiration for the feature originated from the lack of variety in modern news apps. Exploring several apps revealed that most provide a rather limited and boring user experience with little interaction. In order to implement the bonus feature, I will be making use of News API [3] to allow me to pick out key facts, figures and related images to form a bank of questions for each article. I will also use other related articles and calculations performed on correct figures to generate incorrect answers for use in the multiple-choice options.

3.2 Bonus Feature Two: Key Word Article Filtering

In the modern day and age, it is common for news providers to bombard users with certain popular topics. A recent example of this would be Brexit and the Coronavirus. This can often make reading the news repetitive and dull where the user doesn't learn anything interesting. As a result of this common issue, I have decided to implement key word filtering in my app. Under the user profile tab, users will be able to enter keywords (tags) which will appear on screen as toggle buttons in a horizontal scrollable list. There will be 3 scrollable lists, boring, interesting and spotlight, each altering the content displayed on the user's main page:

- **Boring** – Greatly reduced or prevented from appearing on the user's news feed.
- **Interesting** – Will make up the majority of news feed content.
- **Spotlight** – Articles will be placed at the top of the user's news feed for prioritisation over other content.

Tags specified by the user will be used in conjunction with topics chosen in the discover page to filter specific articles which will provide users with a rich and unique news feed. Additionally, due to the use of toggle buttons, users can drastically adjust their news feeds if they wish to be informed of certain developments. An example use case might consist of the following three categorisations:

- **Boring** – An election is due next month, and the user isn't interested in politics and finds the articles misleading. As a result of this, the user specifies: "Trump", "Democrats", "Election", "US", etc.
- **Interesting** – A new technology that interests the user is slowly being adopted by multiple different mobile phone brands. The user wishes to learn about new developments in the industry. Words may include: "Phone", "5G", "Integration", "Samsung", "Apple", etc.
- **Spotlight** – A new Mac Book Pro is being released by Apple, but the release date is not yet confirmed. The user is interested in all articles which contain information of potential release dates. Possible words may include: "Apple", "Mac Book Pro", "Release Date", "Confirmed", "Expected", etc.

Articles which include multiple spotlighted words will be higher in the news feed helping the user to be informed of specific topics as soon as an update is available, as demonstrated above with the Mac Book Pro example.

In addition to considering how the filtering interface will be used by the user, I have also investigated implementation strategies with News API [3]. Observing how news article data is presented by the API, it is clear that the title and description of articles will provide a sufficient insight into the contents of the article and hence its suitability to the user's profile. I will therefore compare each word in the title and description of articles to the user's active tags to generate a match score, based on the number of matched words. From here, the match score will be used to order articles on the news feed.

3.3 Bonus Feature Three: Automatic Scheduled Downloading of Articles

One advantage of Native development is that apps can be run whilst the phone doesn't have an active internet connection, i.e. is offline. I wanted to take advantage of this attribute by implementing automatic downloading of tailored articles for offline use. The user would specify a download and removal time under the profile tab which would instruct the app to save local copies of new tailored articles and delete them at the specified removal time. The user will also be able to specify the quantity of articles that are downloaded and the maximum storage allowance with sliders in the profile section.

An example use case of this feature would be a user who enjoys reading articles whilst commuting to work via the London underground where mobile data coverage is poor. The app could start downloading articles at 07:00 in the morning ready for the commute and remove the articles at 22:00 at night. This offers the potential of maintaining the interactive nature of the app despite that an active internet connection is not available, hence improving the scope of the app.

The inspiration for this feature arose from the lack of clear support for offline mode in several news apps available on the app store. Although the majority of apps offer some level of content whilst offline, most remove images, videos, audio clips, etc. Although it is true that dynamic content like videos and images may have large file sizes, their temporary presence on the phone will not result in large storage overheads. I believe the control that I will provide to the user through the profile section of the app will make this feature stand out from other implementations current on the app store.

4 Interface Design

Throughout the process of designing the user interface, I focused closely on providing a consistent layout and colour scheme. I decided upon a dark colour pallet for my app because it will reduce eye strain due to the reduced brightness and will also save battery life on OLED smart phone screens. I opted for a light blue colour to indicate key points of interest for the user. This will ensure that article metadata clearly stands out from the main article content helping the user to easily focus their time on articles and subjects that interest them the most. I believe that this clarity is crucial to achieving an enjoyable browsing experience.

I decided to position the horizontal scroll views at the top of the screen throughout the app in order to provide a familiar and consistent layout to the user. Additionally, the use of a darker blue underline for activated toggle buttons helps to indicate the customisations that are currently active and affecting on screen content. Positioning the navigation menu at the very bottom of the screen and making use of a vertical scrolling method will ease of operation of app and will support comfortable use with a single hand.

5 Activity GUIs & XML Scripts



Figure 1: Home Recycler View

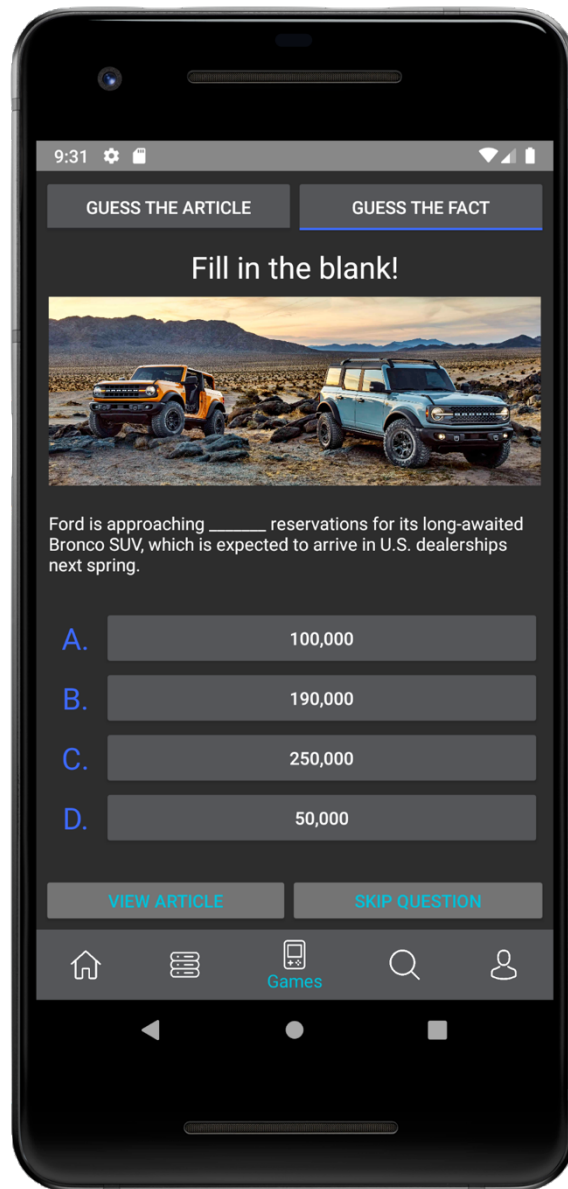


Figure 2: Games

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:clickable="true">

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/combinedCardViewsLinear"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:weightSum="2">

        <androidx.cardview.widget.CardView
            android:id="@+id/smallRowCardViewLeft"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="5dp"
            android:clickable="true"
            app:cardCornerRadius="10dp"
            app:cardBackgroundColor="@color/backgroundGreyMedium">

            <androidx.appcompat.widget.LinearLayoutCompat
                android:id="@+id/smallRowLeftLinearMain"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_margin="10dp"
                android:orientation="vertical">

                <ImageView
                    android:id="@+id/smallRowLeftArticleImage"
                    android:layout_width="wrap_content"
                    android:layout_height="100dp"
                    app:srcCompat="@drawable/dummy_article_image_three"/>

                <ImageView
                    android:id="@+id/smallRowLeftProviderLogo"
                    android:layout_width="match_parent"
                    android:layout_height="20dp"
                    android:layout_marginTop="5dp"
                    android:scaleType="fitStart"
                    app:srcCompat="@drawable/dummy_provider_three_logo_rectangle"/>

                <androidx.appcompat.widget.AppCompatTextView
                    android:id="@+id/smallRowLeftArticleTitle"
                    android:text="@string/dummy_title_three"
                    android:layout_width="fill_parent"
                    android:layout_height="70dp"
                    android:layout_marginTop="5dp"
                    android:textColor="@color/text"
                    android:textSize="14sp"
                    android:textStyle="bold"/>

                <androidx.appcompat.widget.AppCompatTextView
                    android:id="@+id/smallRowLeftCategory"
                    android:text="@string/dummy_category_three"
                    android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="5dp"
                    android:textAlignment="textStart"
                    android:textColor="@color/colorSecondary"
                    android:textSize="12sp"
                    android:textStyle="bold"/>

            </androidx.appcompat.widget.LinearLayoutCompat>
        </androidx.cardview.widget.CardView>
```

Figure 3: Two Column Recycler Row (XML for left Column)

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/backgroundGreyHeavy"
    tools:context=".fragments.GamesFragment">

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/gameSelectorLinear"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginTop="5dp"
        android:orientation="horizontal">

        <androidx.appcompat.widget.AppCompatToggleButton
            android:id="@+id/guessArticleGameToggle"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_weight="1"
            android:text="@string/guess_article"
            android:textColor="@color/text"
            android:textOff="@string/guess_article"
            android:textOn="@string/guess_article"
            android:theme="@style/AppTheme.ToggleButton" />

        <androidx.appcompat.widget.AppCompatToggleButton
            android:id="@+id/guessFactGameToggle"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginRight="5dp"
            android:layout_weight="1"
            android:checked="true"
            android:text="@string/guess_fact"
            android:textOff="@string/guess_fact"
            android:textOn="@string/guess_fact"
            android:theme="@style/AppTheme.ToggleButton" />
    </androidx.appcompat.widget.LinearLayoutCompat>

    <androidx.appcompat.widget.AppCompatImageView
        android:id="@+id/imageView"
        android:layout_width="fill_parent"
        android:layout_height="150dp"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="100dp"
        android:layout_marginRight="10dp"
        android:scaleType="centerCrop"
        android:src="@drawable/ford_bronco" />

    <androidx.appcompat.widget.AppCompatTextView
        android:id="@+id/gameInstructions"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="60dp"
        android:text="@string/guess_fact_instruction"
        android:textAlignment="center"
        android:textColor="@color/text"
        android:textSize="24sp" />

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/multipleChoiceLinear"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/articleContent"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="25dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="65dp"
        android:layout_weight="4"
        android:orientation="vertical">

        <androidx.appcompat.widget.LinearLayoutCompat
            android:id="@+id/choiceALinear"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="horizontal">

            <androidx.appcompat.widget.AppCompatTextView
                android:id="@+id/textA"
                android:layout_width="0dp"
                android:layout_height="fill_parent"
                android:layout_weight="1"
                android:text="@string/a"
                android:theme="@style/AppTheme.SpecialTextView" />

            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/buttonA"
                android:layout_width="wrap_content"
                android:layout_height="fill_parent"
                android:layout_weight="6"
                android:text="@string/dummy_answer_a"
                android:theme="@style/AppTheme.Button.Dark" />
        </androidx.appcompat.widget.LinearLayoutCompat>

        <androidx.appcompat.widget.LinearLayoutCompat
            android:id="@+id/choiceBLinear"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="horizontal">

```

5

Figure 4: Games Fragment XML

```

        <androidx.appcompat.widget.AppCompatTextView
            android:id="@+id/textB"
            android:layout_width="0dp"
            android:layout_height="fill_parent"
            android:layout_weight="1"
            android:text="@string/b"
            android:theme="@style/AppTheme.SpecialTextView" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/buttonB"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_weight="6"
            android:text="@string/dummy_answer_b"
            android:theme="@style/AppTheme.Button.Dark" />
    </androidx.appcompat.widget.LinearLayoutCompat>

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/choiceCLinear"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="horizontal">

        <androidx.appcompat.widget.AppCompatTextView
            android:id="@+id/textC"
            android:layout_width="0dp"
            android:layout_height="fill_parent"
            android:layout_weight="1"
            android:text="@string/c"
            android:theme="@style/AppTheme.SpecialTextView" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/buttonC"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_weight="6"
            android:text="@string/dummy_answer_c"
            android:theme="@style/AppTheme.Button.Dark" />
    </androidx.appcompat.widget.LinearLayoutCompat>

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/choiceDLinear"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="horizontal">

        <androidx.appcompat.widget.AppCompatTextView
            android:id="@+id/textD"
            android:layout_width="0dp"
            android:layout_height="fill_parent"
            android:layout_weight="1"
            android:text="@string/d"
            android:theme="@style/AppTheme.SpecialTextView" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/buttonD"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_weight="6"
            android:text="@string/dummy_answer_d"
            android:theme="@style/AppTheme.Button.Dark" />
    </androidx.appcompat.widget.LinearLayoutCompat>

    <androidx.appcompat.widget.LinearLayoutCompat
        android:id="@+id/navigationLinear"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:layout_alignParentBottom="true"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="3dp"
        android:gravity="bottom"
        android:orientation="horizontal">

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/viewArticleButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/view_article_button"
            android:theme="@style/AppTheme.Button.Nav" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/skipQuestionButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/skip_button"
            android:theme="@style/AppTheme.Button.Nav" />
    </androidx.appcompat.widget.LinearLayoutCompat>

    <androidx.appcompat.widget.AppCompatTextView
        android:id="@+id/articleContent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/imageView"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="10dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="10dp"
        android:text="@string/dummy_question"
        android:textColor="@color/text" />
</RelativeLayout>

```

Figure 5: Games Fragment XML (Continued)

6 Works Cited

- [1] The Mirror, [Online]. Available: <https://www.mirror.co.uk>. [Accessed 28 October 2020].
- [2] The Sun, "Home," [Online]. Available: <https://www.thesun.co.uk>. [Accessed 29 October 2020].
- [3] News API, [Online]. Available: <https://newsapi.org>. [Accessed 28 October 2020].
- [4] Quillionz, [Online]. Available: <https://www.quillionz.com>. [Accessed 27 October 2020].