Description

Intended User

Features

User Interface Mocks

News Screen

Article Details Screen

Squad Screen

Player Details Screen

Fixtures Screen

App Widget

Key Considerations

How will your app handle data persistence?

Describe any edge or corner cases in the UX.

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

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

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Set up Google Cloud Endpoint server

Task 3: Create News and Article details activities

Task 4: Create Squad and Player details activities

Task 5: Create Fixtures and Results activity

Task 6: Synchronize fixtures with calendar

Task 7: Implement push notifications

Task 8: Add App Widget

GitHub Username: Norbi930523

Manchester United App

Description

This application aims to deliver up-to-date information to its users including fixtures, results, squad and the latest news of Manchester United football club. The source of these pieces of information is the official website of Manchester United (http://www.manutd.com/), therefore the application is not intended for commercial use.

Intended User

The target audience of this application are the fans of Manchester United football club.

Features

- Latest news: read the latest news about Manchester United
- Squad list: check out the most essential information about the current squad
- Fixtures and results: check out the upcoming fixtures and the team's results
- Calendar integration: synchronize the team's fixtures with your calendar and set reminders
- Push notifications: get notifications about the Starting lineup for the next match and goals
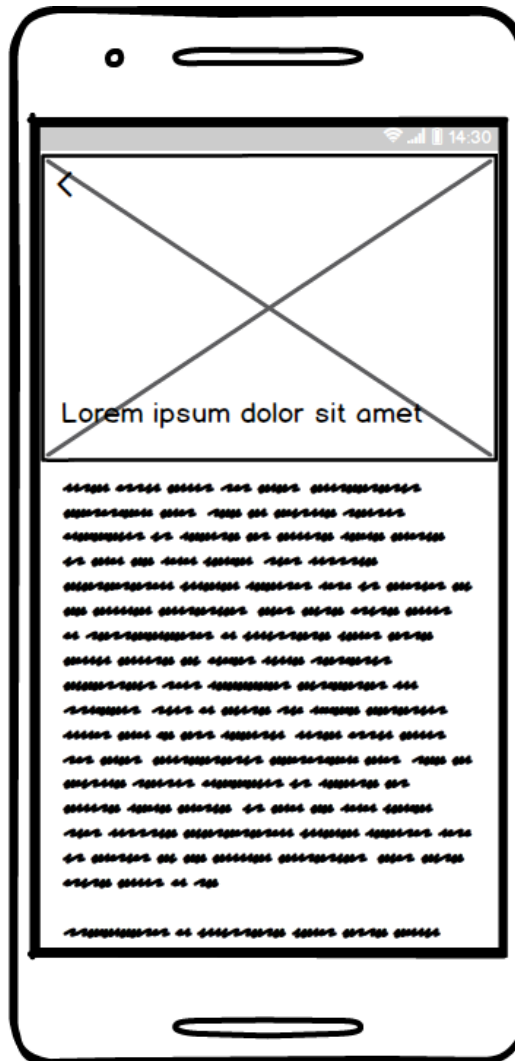- App widget: shows the date of the upcoming match
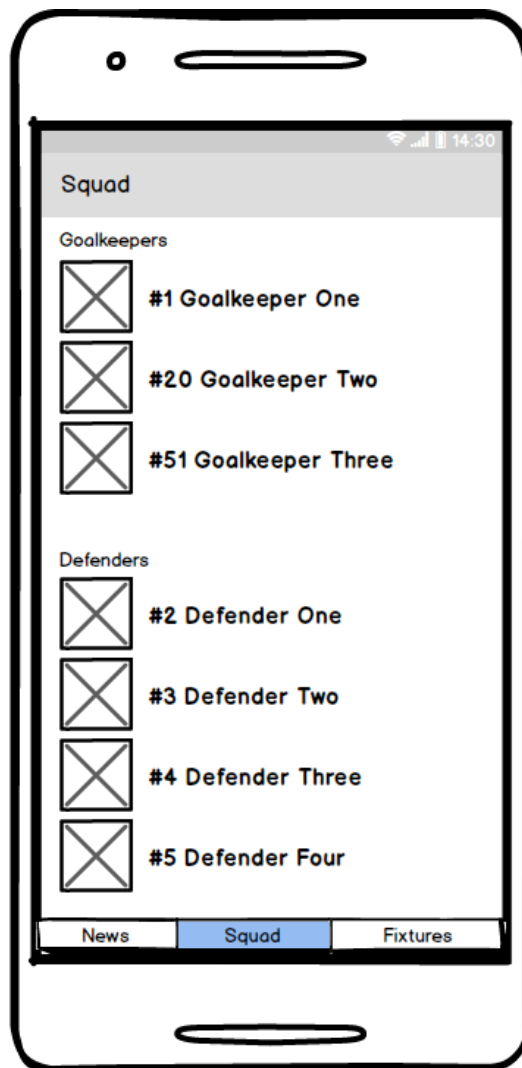
User Interface Mocks

News Screen
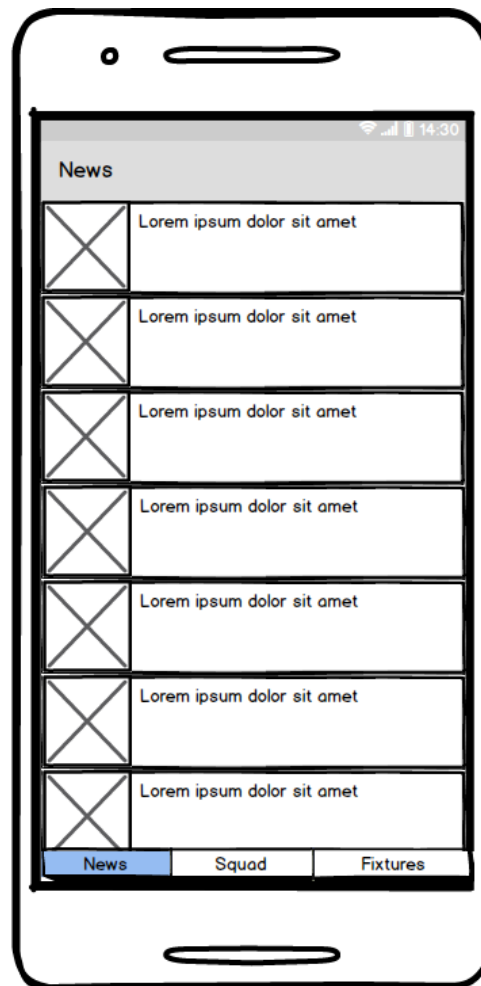
The user starts here when the app launches. The screen shows a list of articles, each with an image and a title. The user can navigate through activities via the Bottom Navigation View.
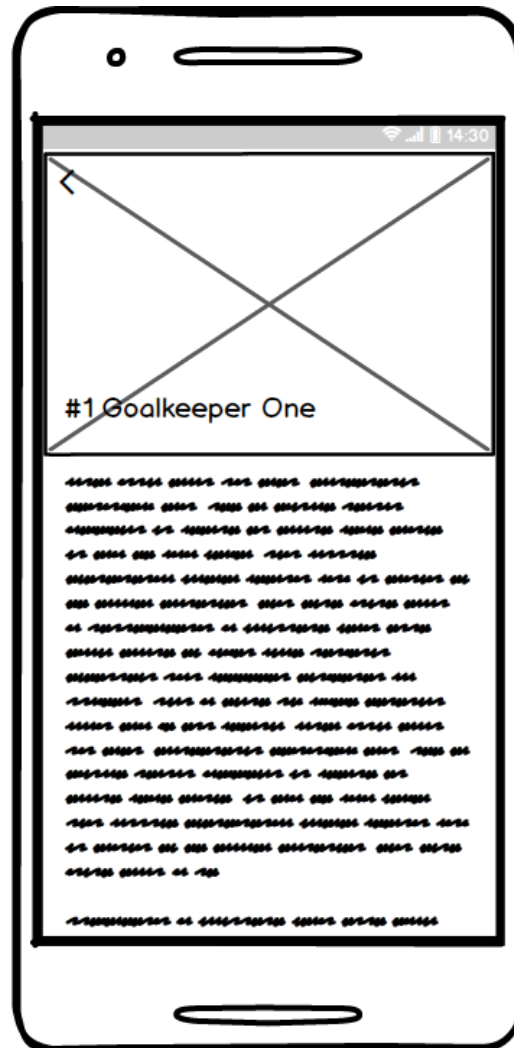
Article Details Screen

The screen shows the selected article's image, title and content. The image fades away as the user scrolls down.
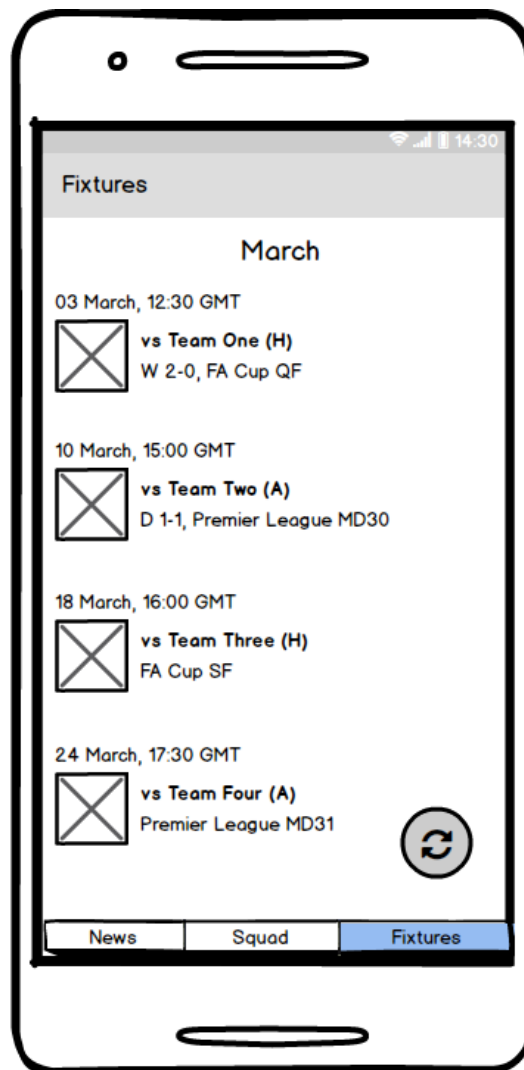
Squad Screen

The screen shows the list of players with their image and squad number, grouped by position. Clicking on an item navigates the user to the player details screen.

Player Details Screen

#1 Goalkeeper One

The screen shows the image, name and details of the selected player. The image fades away as the user scrolls down.

Fixtures Screen

The screen shows the list of matches for a given month with the exact date, the opponent's name and logo and the competition. The user can page through the months by swiping left or right. The user can synchronize the fixtures to the calendar using the Floating Action Button.

App Widget

Next match
07 April, 17:30 BST
vs Team One
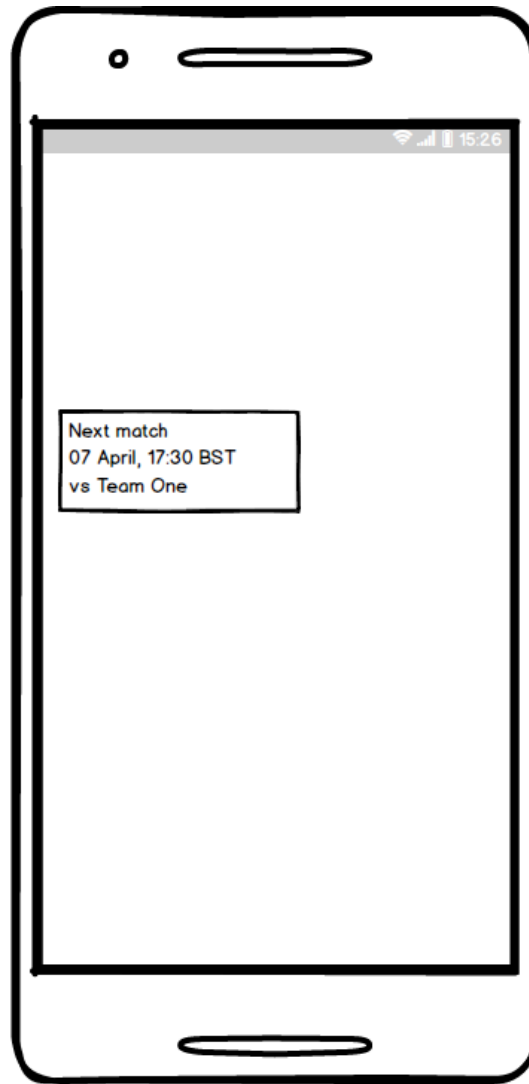
Shows the date of the upcoming match.

Key Considerations

How will your app handle data persistence?

The application will fetch the data from a Google Cloud Endpoint server through an IntentService and cache the data in a local SQLite database using a ContentProvider.

Describe any edge or corner cases in the UX.

Since the application will mostly use a list – detail flow for navigation, which is not a best practice for tablets, it would be best to use a master – detail pattern to be able to provide the best user experience for both phones and tablets.

Users will be able to synchronize fixtures to the calendar using a Floating Action Button.

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

- **Timber:** to make logging easier
- **Schematic:** to create ContentProviders easily
- **Picasso:** to load images of articles, squad members and logos of other teams
- **GSON:** to parse JSONs easily
- **Google API and HTTP client:** to communicate with the GCE server easily
- **Android Design Support library:** to provide a better user experience
- **Android AppCompat library:** to support a wider range of devices

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

The application will fetch data from a Google Cloud Endpoint server. The following API endpoints will be implemented:
- News, squad, fixtures: returns the contents of a corresponding local JSON file, which will contain data downloaded from manutd.com
- Register / unregister GCM tokens
- Push notifications for match events (starting lineup, goals): users will be able to trigger match events (like starting lineup announcement, goals scored) using a simple user interface and send push notifications to devices using the Google Cloud Messaging service

Next Steps: Required Tasks

Task 1: Project Setup

1. Create new Android Studio project
2. Create Signing Configurations
3. Add dependencies: Android Design support library, Google API client, Google HTTP client, Picasso, Timber
4. Set up logging with Timber

Task 2: Set up Google Cloud Endpoint server

1. Add new Google Cloud module
2. Configure dependencies for module
3. Create local JSON files for news, squad and fixtures
4. Create and implement API endpoints to get news, squad and fixtures

Task 3: Load and store data from server

1. Create an IntentService to load data from the server
2. Create an SQLite database and a ContentProvider to be able to store the response from the server
3. Extend the IntentService with storing the server response in the database using the ContentProvider

Task 3: Create News and Article details activities

1. Create new Activity to show the latest news
2. Load articles from the corresponding API endpoint using the IntentService
3. Show articles in a RecyclerView via a CursorLoader
4. Create new Activity to show a single article in detail
5. Add a click listener to the article list items to show the Article details activity

Task 4: Create Squad and Player details activities

1. Create new Activity to show the list of players
2. Load squad from the corresponding API endpoint using the IntentService
3. Show players in a RecyclerView via a CursorLoader
4. Create new Activity to show a single player in detail
5. Add a click listener to the player list items to show the Player details activity

Task 5: Create Fixtures and Results activity

1. Create new Activity to show the fixtures and results of the team
2. Load fixtures and results from the corresponding API endpoint using the IntentService
3. Show fixtures and results in RecyclerView via a CursorLoader
4. Create button to synchronize fixtures to calendar

Task 6: Synchronize fixtures with calendar

1. Add WRITE_CALENDAR permission to AndroidManifest to be able to write to calendar

2. Synchronize fixtures to calendar using the Calendar Provider

Task 7: Implement push notifications

1. Configure Google Cloud Messaging
2. Create an API endpoint for managing GCM tokens
3. When the user starts the application, register GCM token on the server using the API endpoint
4. Create BroadcastReceiver to be able to handle push notifications coming from the server
5. Create API endpoints for triggering push notifications from the server

Task 8: Add App Widget

1. Create WidgetProvider with widget info XML
2. Create a layout for the widget
3. Implement widget updates in the IntentService