CA326 THIRD YEAR PROJECT

AUTHORS:

KARL DOHERTY: 19413086

YURY CHUPAHIN: 19416764

Project Name: PremNews

SUPERVISOR:

MARK ROANTREE

Table Of Contents

1. Introduction

1.1 Abstract

Our project, PremNews, is an android application that features news articles from multiple sources which are scraped from the internet, uploaded to a real-time cloud database and run remotely on a schedule using google cloud functions[1] and google cloud scheduler[2], and designed on the front end using android studio. The scrapers are written using python, the database provider is firebase[3] and Java and XML are used as part of the front end design.

1.2 Context/Background

The idea for our project, PremNews, initially came from an experience we had during the summer of 2021. During the summer months, the transfer window opens, allowing teams to buy and sell players. It is one of the most interesting and dramatic times of the year as a football supporter (excluding the games themselves) as your team could end up selling important players, or strengthening the squad with more talent from other teams. Every football fan is intently trying to stay updated on the latest news, including myself and Yury. We found ourselves going from site to site looking for news and wasting much time navigating between links until we found what we were looking for. This is where the original idea from this project came from as we thought it would be convenient to have a multi-source news app for up to date football news.

This app is designed to provide users with a convenient way of accessing news articles from multiple sources all in one place, as well as the ability to filter news articles by date, team and source. Users of this application are able to log in, choose their desired teams and only see news from those teams, or only see news from a specific date if they desire. Most large scale news providers do not have such specific filtering available and sometimes users are given back a large amount of news articles to scroll through, some of which are completely unrelated to the topic the user was originally searching for. This causes users to waste time searching through unrelated topics for news instead of being able to find it quickly and concisely in one easy to use application.

1.3 Product and Technical Requirements

1. As a fan of the premier league, I like to keep up to date with all the news that is going on and be able to see different articles and stories all in one place. This allows me to see the different opinions of different writers on the same topic
2. As a user of this app, I want to be able to set a couple of teams that I have an interest in, and only see news from those teams. This will allow me to not waste time looking at news that I have no interest in.

1.4 Glossary

A
Android Studio - the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

F
Firebase Realtime Database - The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in realtime.

G
Google Cloud Functions - Cloud Functions allows users to run your code from Google Cloud, Firebase, and Google Assistant, or call it directly from any web, mobile, or backend application via HTTP.

Google Cloud Scheduler - Cloud Scheduler is a fully managed enterprise-grade cron job scheduler.

J
Java - Java is a high-level, class-based, object-oriented programming language

P
Python - Python is a high-level general-purpose programming language.

W
Web Scraping - Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser.

X
XML - XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use.

2. General Descriptions

2.1 Goals

Our goals are:

- Design an application that takes in news from multiple sources.
- Stores this news in a real-time cloud database, which is structured and pre-processed to allow for easier filtering on the front end.
- Create scrapers that are run remotely by cloud functions to continuously add data
- An Android Studio frontend application that displays these news stories and allows users to register and filter data based on their favourite teams.

2.2 Future Goals

If we were to continue working on this application, there are an endless number of features that we could implement to continue to broaden the horizons of PremNews. We could begin to add more sources for news and start taking in news from other leagues and countries and start filtering the database even more. The user interface could be improved even more to help modernise the application even further and present users with an even more enjoyable experience when using the app. Eventually, it could end up being a one-stop-shop for any kind of football news, possibly even sports news in general, with very specific filtering options and constant up to date information due to the constantly run scrapers on the cloud.

2.3 Non-Goals

As mentioned in the future goals section of the specification, in the future it would be possible to develop this application into a one-stop-shop for all kinds of sports news from multiple countries. However, given the timeframe of the project, it was not feasible to complete such a project of that size, so we had to make some sacrifices and focus on the Premier League as the source of news we would provide to users.

2.4 Requirements

ID	Requirement	Description	Criticality
1	App Development Environment	A software where applications can be created and customised. This is Android Studio in our case	Critical
2	A Realtime Database	A realtime database that allows news stories to be stored	Critical
3	A remote method of updating the database	A method of allowing the scrapers to run remotely. We used Google Cloud Functions	Important
4	A user interface	An easy to use user interface that users find simple	Important
5	Programming Languages	Java, Python, and XML	Critical

2.5 Assumptions

- Both members will have access to computers
- Both members will have Android Studio installed
- Both members will be able to access the internet

3. Solution

3.1 Suggested/Proposed Solution/Design

Our initial idea for this application was to take opinions from sports journalists on social media apps like Twitter and Facebook and put these along with the scores in games as a way of showing people's opinions on the game. We sided against this as we thought it would be too large of scope to try to create an application of that complexity. It would be very complex to try to distinguish between positive things about teams and negative, as well as profanity and biased tweets. This would have been an interesting idea had we had more time to research the subject but we sided against it as we wanted to make sure that whatever project we choose to build would be completable in the timeframe.

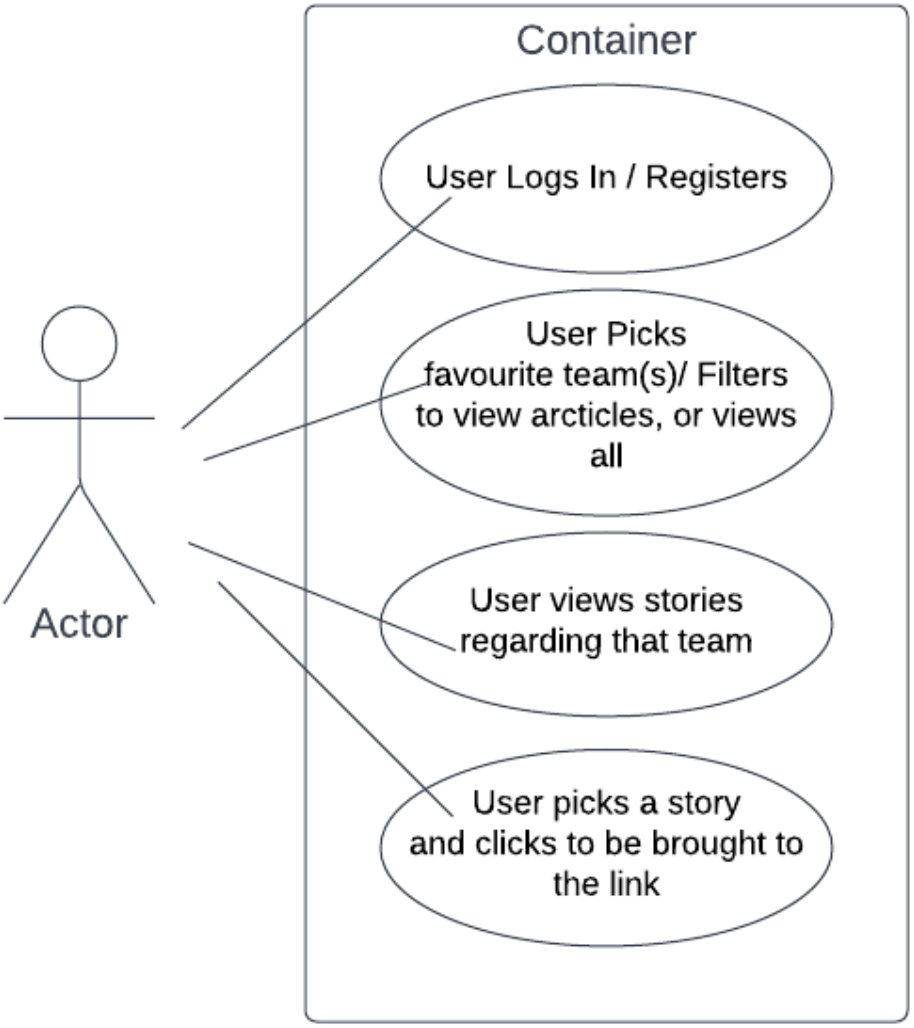
3.2 Current/Existing Solution/Design

Our design for the current solution is a simple but effective solution that is user friendly. When the application is opened, users are prompted to log in or register if it is their first time using the application. After logging into the app, users can then pick their preferred teams they would like to see news from. They also have the option to only see news from a particular source or date.
The design of the backend is a series of scrapers written in python. The scrapers are designed to run remotely on the cloud every two hours to constantly keep an up to date stream of stories being added to the database. Each element in the database contains four attributes:

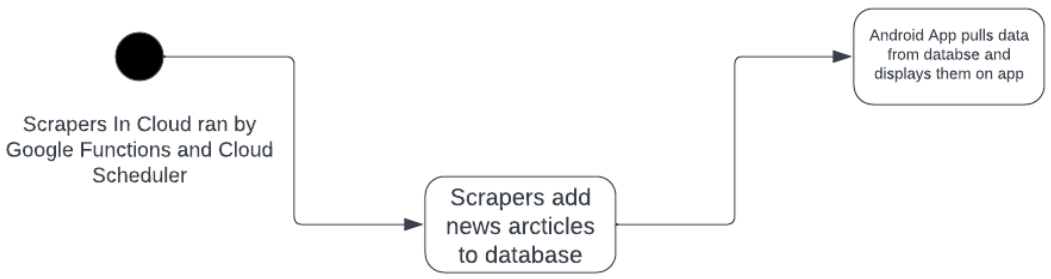
- Title
- Link
- Source
- Date Added

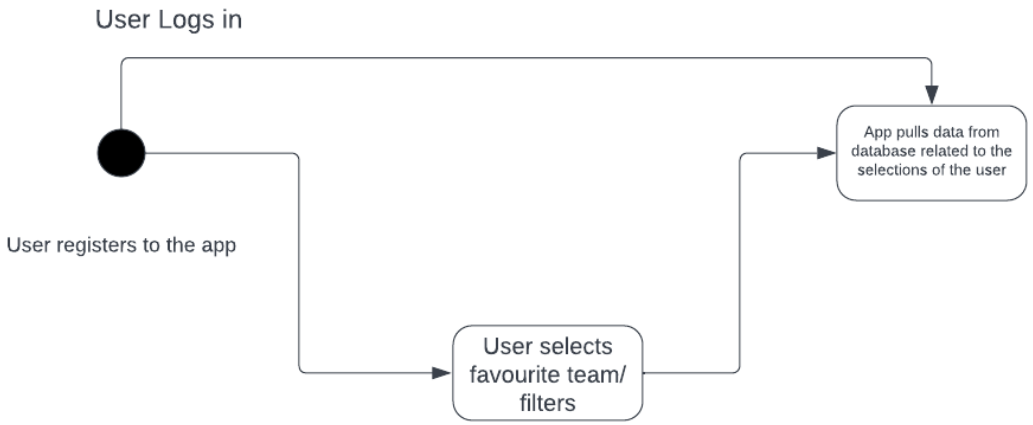
This preprocessing of the data allows the filtering of the data to be a lot easier and less time consuming, as Android Studio was a new technology to us. Users can log in and their favourite teams are stored in a database for when they log back in. The app features a home page showing all the filtering options, along with information buttons users can click to see actually what the button they are clicking does. The articles are laid out in an easy to read list that shows the user the date added as well as the title, which is clickable and brings the user straight to the article or page of information they click on. The user can then back out and go back to the list of articles and view another article, or narrow their search down with a keyword search, or the filters provided. The user can then leave the app or log out and their preferences on teams will be saved for them the next time they log into the app, saving them some time

3.3 Use Case Diagram



3.5 State Diagrams





4. Testing

4.1 How We Plan To Test Our App

Unit and Integration testing are both carried out during the development stage. During the development of the application, we will constantly test the features of the application to ensure it was working correctly. Unit testing was conducted during the early stages of development to ensure specific parts of the project are working as intended. While the scrapers run, a message is logged whether the article already exists in the database or it is currently being added, as well as a count of how many articles have been added currently. While designing the scrapers, multiple tests were run on the scrapers to ensure that they were scraping the data correctly and pushing the data to the database correctly, as well as tests to check that new data was being added every day and the cloud functions were running on schedule.

```
Already exists. Ignoring ...
Already exists. Ignoring ...
Already exists. Ignoring ...
The scrapers are currently working correctly.
The database currently contains 435 news arcticles from 4 sources
PS C:\Users\Vassi\Desktop\CA326\2022-ca326-doherk27-chupahy2\src\scrapers>
```

This is an example of the program returning the health of the scrapers as well as the number of articles present.
On the front end, the code was constantly written to log all responses, making it easier to identify and rectify bugs in the code quicker.
Integration testing will be carried out throughout the development of the project to ensure all components of the application are working.
The application will be tested on a physical Android phone as soon as a working prototype is produced, and news articles will be visited and filtered from the database.
The application will be sent to peers and colleagues for testing and these individuals will be asked to give feedback based on their thoughts. This helped us identify a couple of bugs that we did not find ourselves and helped us to improve the application even further.

5. Deployment Plan

Planning is essential in any large project, especially a team one. As a team, we are implementing a type of scrum sprint cycle, each week we will set out goals for the week and assign each other an equal workload, with Karl focusing on the backend and Yury working on the front end. We will meet regularly and discuss our progress, methods, opinions and plans going forward. This strategy is beneficial for several reasons. It keeps everyone working to their full potential as there is constant peer review, and it gives the developer a clear goal to work towards and a next step in putting the project together.
We constantly update the code on GitHub, along with constantly keeping each other in the loop about any changes or problems in the code, which helps prevent wasting time as we both know what stage the other was at.

6. Successful Evaluation

The goals we set for ourselves and the project at the beginning were to develop an android application that takes in news about the Premier League and adds it to a database so it can be taken in on the front end and displayed in a user-friendly way. We planned for the code to be run remotely but were unsure of the method how previous to doing some research on the topic. We planned to allow the app to have multiple different filtering options including keyword, date, source and team filters.
We wanted the user to be able to find their news quickly and without having to scroll through unwanted articles.
We believe that we were able to achieve all these goals we set at the beginning of the project, and evaluating ourselves now we are satisfied we have been successful in completing the challenges we set ourselves.

7. Acknowledgements

This project has involved many people besides the creators of this application, and could not have been possible without the help of many outside sources. We consulted with many friends, colleagues and people who have an interest in the topic we have chosen to build our app on. These people provided us with valuable information and ideas which we could implement into our app, as well as use cases for our app and help with the design of the UI.

We also gave some of our friends the opportunity to test the app at different stages of development. We would like to thank those people for their honest opinions and valuable suggestions.

Finally, we would like to thank the supervisor of this project, Mark Roantree for his guidance and help during this project. Mark helped steer us in the right direction from start to finish, and for that we are grateful.

8. References

[1] **Google Cloud Functions Documentation**
https://cloud.google.com/functions/docs

[2] **Google Cloud Scheduler Documentation**
https://cloud.google.com/scheduler/docs

[2] **Firebase Documentation**
https://firebase.google.com/docs?gclid=Cj0KCQiA64GRBhCZARIsAHOLriLxgxkDrtuCnwSgYfy2EZDg5ZtaTLwi_t36wybVl7yYTWR4c2OtA4gaAukHEALw_wcB&gclsrc=aw.ds