**Product Design**

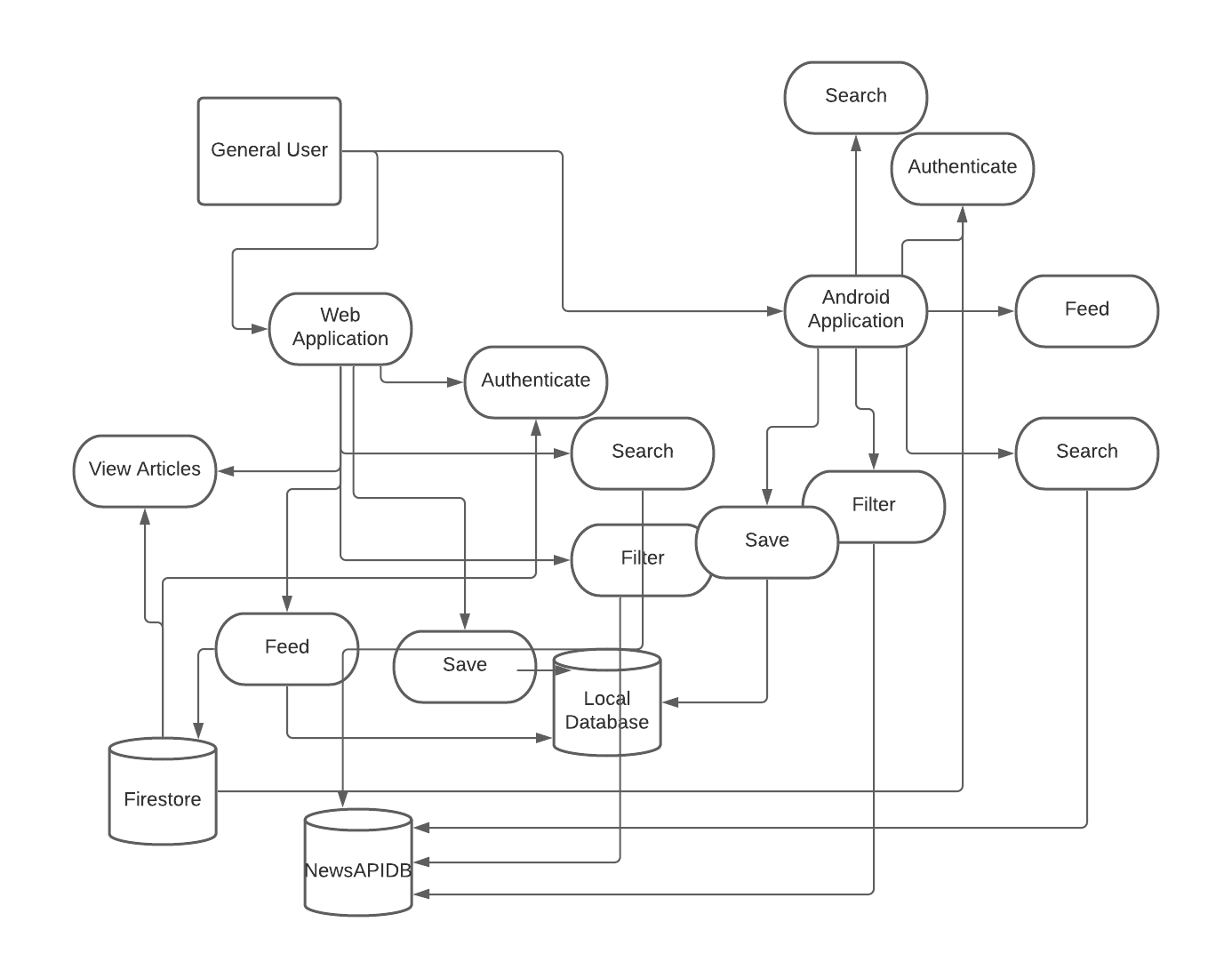
|  |  |
| --- | --- |
| **Team** | **<15, PicNews, Trinadh Venkata Satyanarayana Dusanapudi, Shivanshu Jain, Srijith P,Varul Srivastava>** |

# Design Overview

## Architectural design

The Picnews Application developed by us consists of several modules working together to achieve our desired result of creating an application that displays news articles in a form that saves time of the reader.

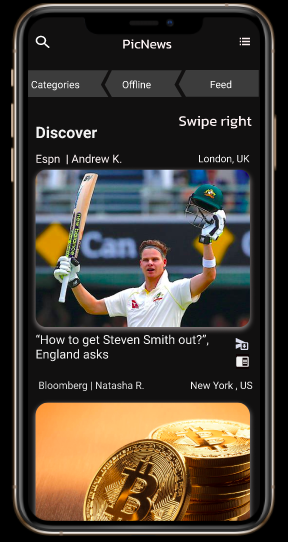
Our system is broadly subdivided into Database, ML component, local database, android application, web application. And web is divided into search portion, shared link, filter news articles, view news. Android is divided into search , filter, view news articles, feed components.



## System interfaces

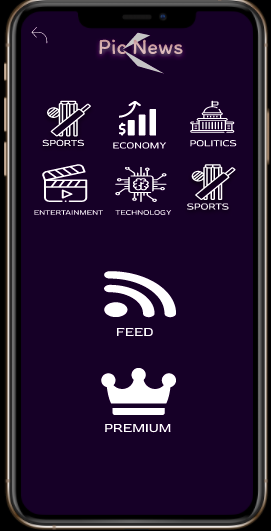
### User Interface

***Android APP Interface:***



***HOME SCREEN***

*This will be our home screen, our users will be directed to this page wen they login. We have various options like search, categories, offline and feed each of these pages screen shots are given below. As far as features are concerened search will redirect to the page where the user can search, categoies will redirect to a page where there are different categories. And offline and feed will refirect to the pages in which the articles are displayed which are downloaded , based on the history of the user respectively.*



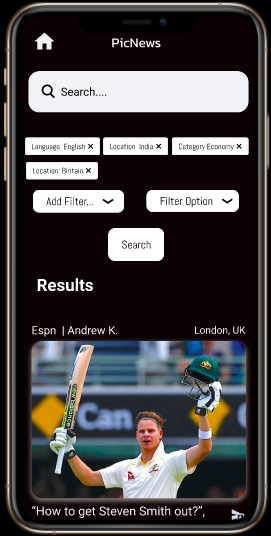
***Categories Page***

*This is categories page, where the user can find wide range of categories including but not limite to sport, entertainment, politics, feed, premium. Our user can view respective categories just but taping them. They can also go back to the homepage by clicking on the arrow button on top left.*



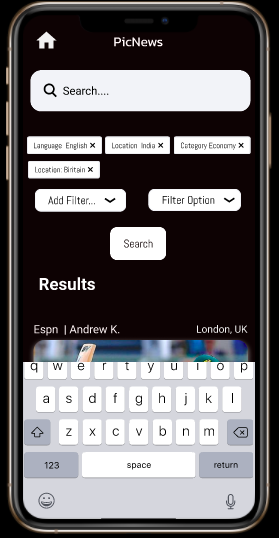
***Article***

This is how the user will be presented the news **artile WOW**  when tapped in the article image in the home screen. As you can see in the screen shot above user will be presented with share and save article as shown in top right upon clicking the user can share, save the article respectively.



***Search and Filter Page***

***User*** *can search for articles by typing in the test box as shown above, upon clicking on the text box user will be presented with onscreen keyboard as shown below. User can also filter based on tags like country, location and many more. Upon clicking search user will be presented with corresponding articles in the same page.*



**WebAPP Interface**

 ***Home Page***

User can login to page using the login option provided on top right, user can also register using the register option. User can view the content by category by selecting the options given in the see news by category towards the bottom left of the screen. User can also search for articles using the search option and user can filter news articles on tags like country, language etc..

### APIs

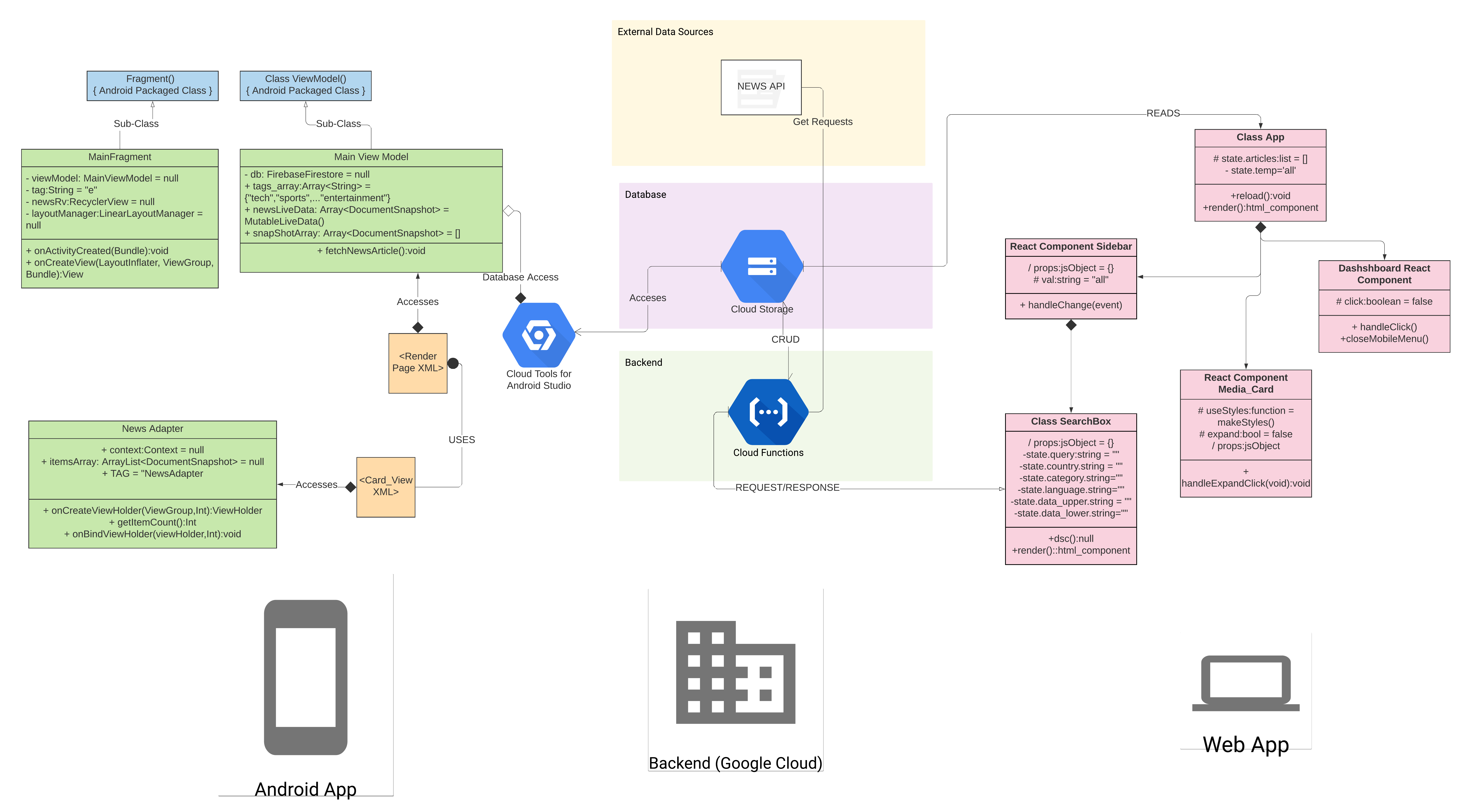
**1. News API :** This api is the core to the data we provide. It is an online support that provides news articles based on various filters and search queries. We make scheduled api calls to news api via firebase cloud functions and store the data in the backend using firebase firestore.

**2. Firebase(Node.js) API :** This api allows us to link our react js based web application with the backend to make firebase function calls and apply crud options on firestore database

**3. Firebase(Android)API:** This api allows us to link our android application with the backend to make firebase function calls and apply crud options on firestore database.

**2. search Query :** Firebase implemented function that takes query and provides search results for relevant news articles.

## Model



|  |  |
| --- | --- |
| <Main Fragment> | eClass state   * Class is responsible to render data and functionality from other components to the XML portion (UI) of the Android Application   Class behavior   * Creates Views, Activities, Links to XML Components using functions implemented within, over-riden from parent class as inherited from parent class. |
| <Main View Model> | Class state   * Uses Firebase API for Android Apps to access database and loads into local Database   Class behavior   * fetchNewsArticle() method imports news articles from cloud firestore based on category and loads them to local Database (for speed) |
| <News Adapter> | Class state   * Renders components to the Render news Page of the App   Class behavior   * onCreateViewHolder() creates an instance of viewHolder class with all components * onBindViewHolder() binds them and gives them functionality while binding to XML page |
| <App> | Class state   * Manages navigation and renders html for the entire web application. Also fetches data from firestore   Class behavior   * render() manages all html component and interlinking of different components * reload() and constructor are for their normal usage as per the application’s logic. |
| <SideBar> | Class state   * Implements 1. Show news by category 2. Superclass for search functionality   Class behavior   * It is a react Hook and it shows news articles based on the category selected within the radio button. |
| <SearchBox> | Class state   * Implements the search news functionality via cloud firebase function calls.   Class behavior   * react hook that searches according to the filters and data provided. Makes use of Axios and Firebase Function calls for the same. |
| <DashBoard> | Class state   * Implements Dashboard funcitonality   Class behavior   * Acts as link between pages for login and signup and the main web app page. |
| <Media\_Card> | Class state   * Shows individual news articles as single UI component. Implements functionalities as well   Class behavior   * Implements Functionality for opening links to source, share news, and show entire article or a small part of it. |

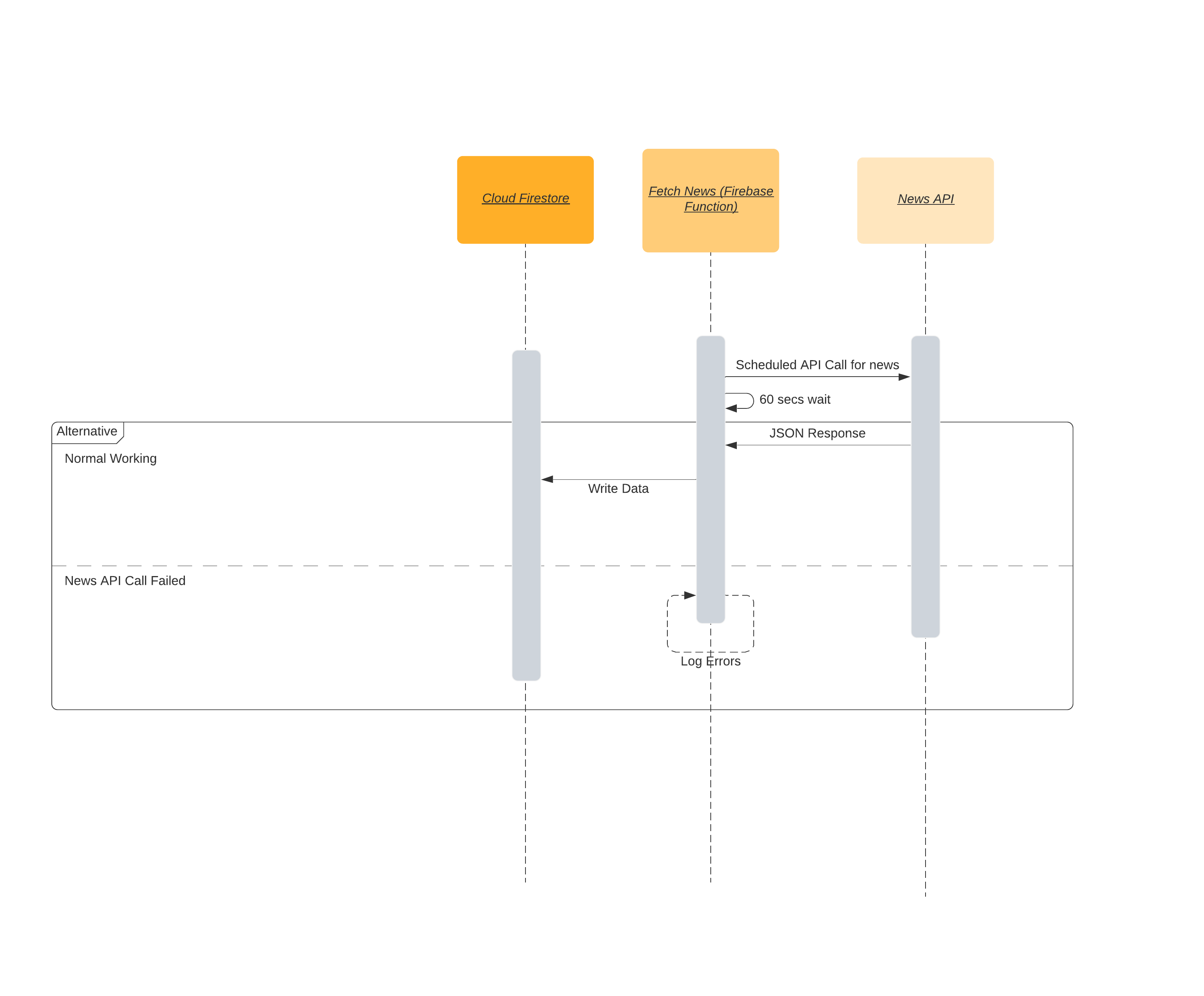
.

# Sequence Diagram(s)

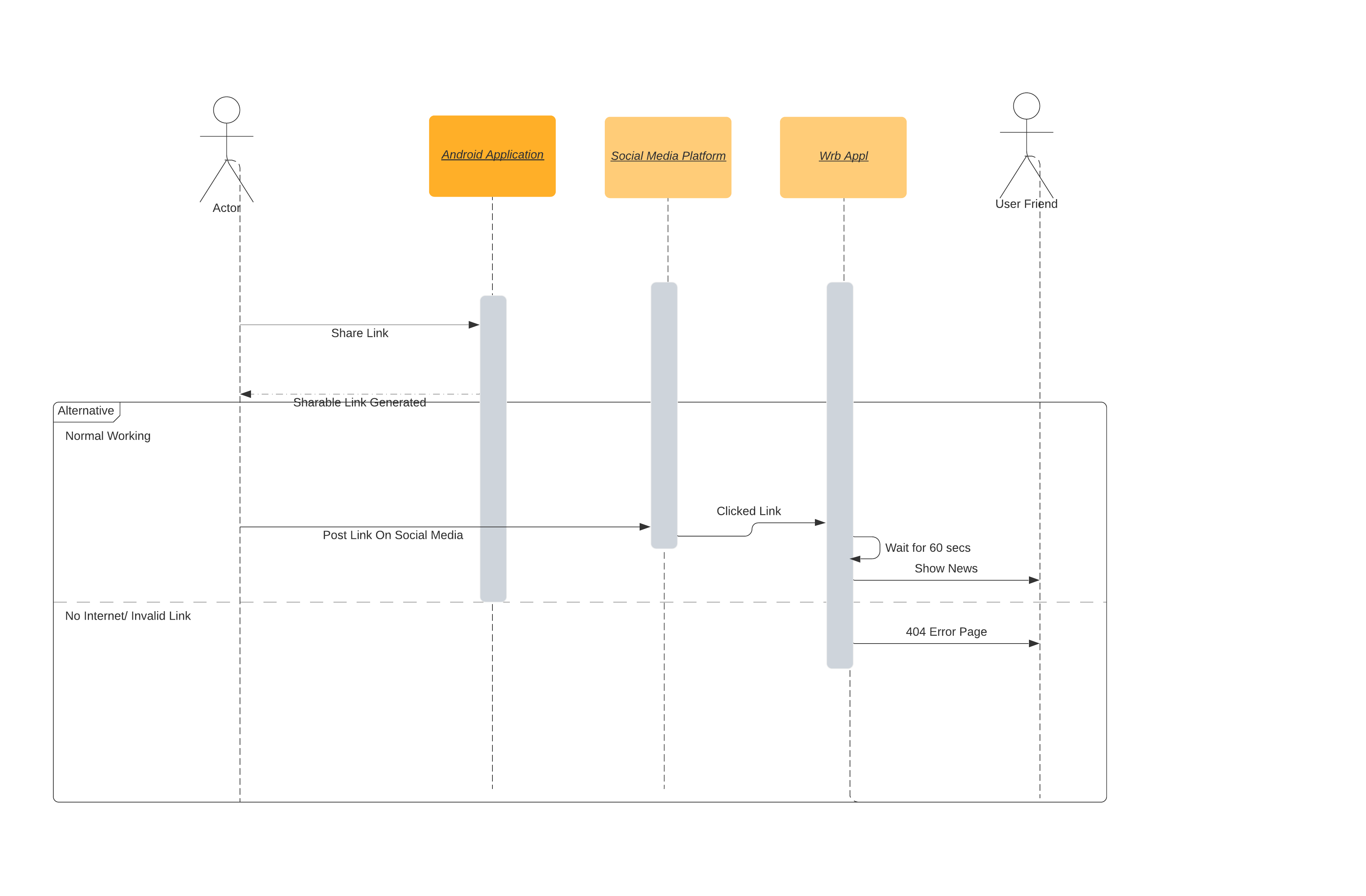
1. Save A Story

# 

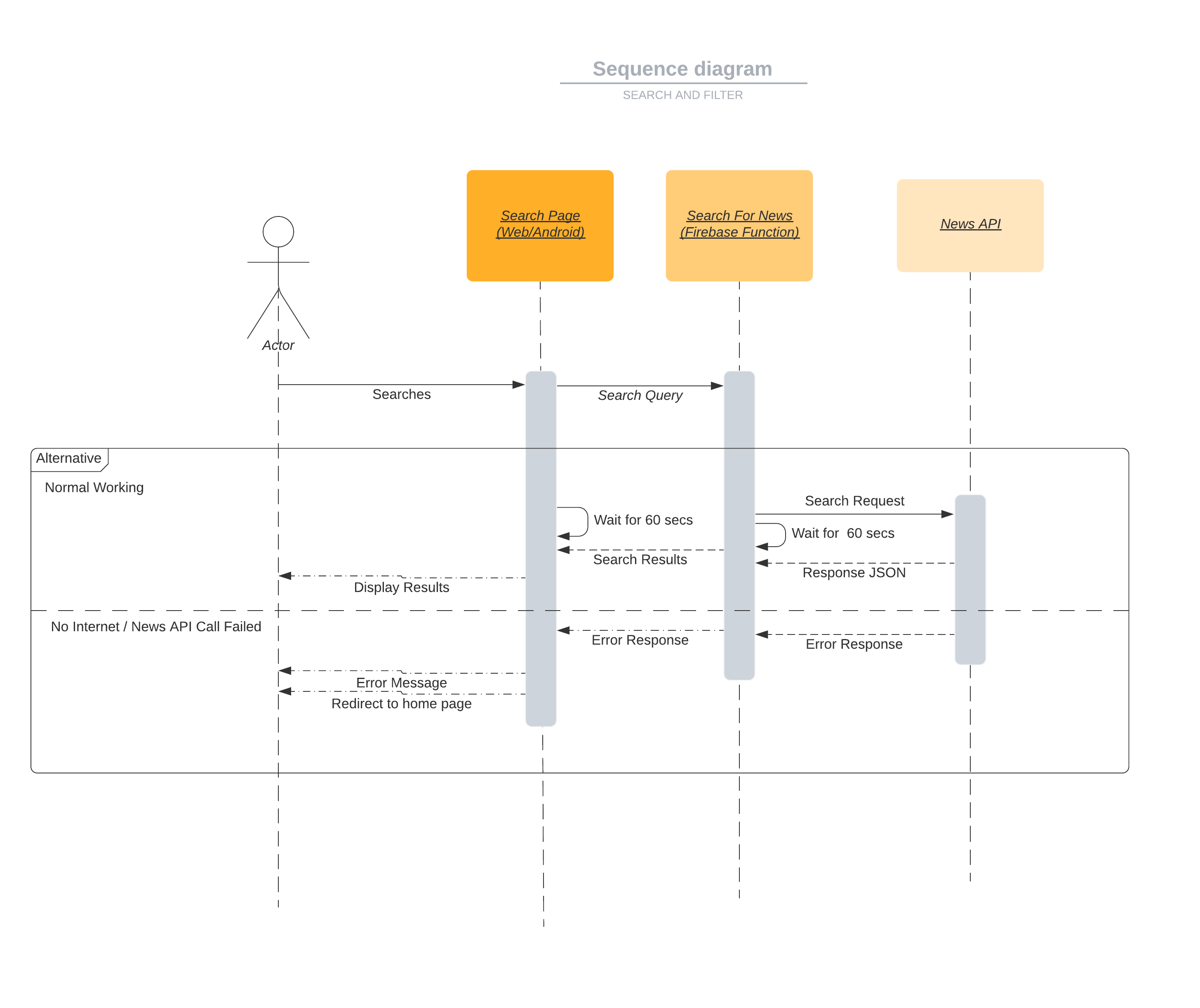
1. Fetch News Articles to Database



1. Share News



1. Search and Filter



# Design Rationale

**1. Backend Database Schema**

**Issue :** Database Optimization. The database schema that we came up with for optimization had the following issues : 1. Used more complex model and required more time to write. 2. Required more time to yield results based on certain filters.

**Options tried :** We tried to correct the schema through multiple alternatives but always got caught up in the problem of slowing down the more frequent processes by speeding up rare ones.

**Solution :** We decided to implement a single layered database on firestore, that will exploit search functionalities built in. We decided to move any potential optimization to the frontend i.e. the Android Application.

***2.* React JS web-app to Cloud Backend calls**

**Issue :** By default cors is disabled in firebase functions, so we cannot make http requests to backend functions and fetch database.

**Options Tried :** Tried multiple options as allowing anonymous cors, adding web-app to the list of known urls etc. But all led to problems in making requests to backend, or were inefficient as they couldnot access firestore database directly.

**Solution :** Made use of Firebase provided API to make calls to firebase functions and access firestore directly (without any middle-functions). Also opens up option for google login authentications, ads and analytics apis to be used in future, along with increase in speed of request/response.