CS307 – Software Engineering
Design Document

NewsLog:
Filtered News According to User Interest

Team 4, Members:

Paul Heldring
Joe Nathan
Austin Dewey
Stephen Motherwell
Akshit Kandi
Dhruv Subramanian
Misha Malik

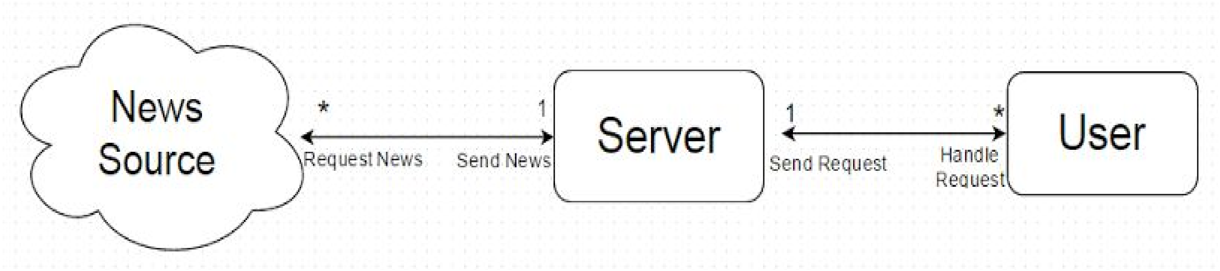
TABLE OF CONTENTS

Purpose

Today, there are many news sites and applications that provide news based on categories and topics, but few provide an adaptive news experience that change the news articles presented based on users likes and dislikes. The purpose of our project is to acquaint the user with a broad range of topics based on certain categories, and help specify their news log based on previous articles they find interesting.

Figure 1.1: High level overview of our system

All of our users will be connected to our central server, which processes requests sent by the user. These requests include updating the user's account information, requesting more articles for the user's news feed, and requesting the user's saved article library. The server then handles the request by either executing the user's desired function or requesting more news articles from the application's news source.



The following items outline the layer to layer interactions and expectations that are necessary for proper functionality of our application.

- **User Account Management:**

  Applications users must be able to manage their account information according to options provided by the developer. A user will be able to select news categories and switch between categories according to their interests. The user will also be allowed to manipulate the algorithmic level feed by having the power to directly manipulate their provided news articles based on their likes and dislikes. The user must also be given the option to view and manipulate personal account information.

- As a user, I want

    - To be able to create an account
    - To be able to delete my account
    - To be able to change my password
    - To be able to safely recover my account password
    - To be able to remain signed in when I close the app
    - To be able to login using my email address.

- **News Interactions:**

  The user will select a personal news channel from a predefined list of interests. The application will populate a list of news channels to be presented to the user. The user will have the option to like or dislike these articles and directly customize their feed based on a variable that tracks their likes and dislikes. The news feed will update based on the initial topic selection; but as the user customizes their feed, the feed will consider these selections during future database article pulls.

    - As a user, I want (include what the feed can do for the user)

        - To search news based on keywords (e.g. Lakers, Apple, etc.)
        - To filter news based on category (e.g. Politics, Tech, etc.)
        - To be able to reset my interests.
        - To be presented articles based on my likes and dislikes of other articles
        - To swipe left/right based on my disapproval/approval of an individual article presented to me
        - To save an article to read it later
        - To have a library of articles that I've saved
        - To be able to clear my library of saved articles
        - To be able to read an entire article immediately when it is presented to me
        - To switch between news feeds based on category

    - As a developer, I want (include broad details about feed functionality)

        - To deliver accurate news articles to the user
        - To limit the maximum size of a user’s library
        - To receive user feedback
        - To be able to delete a user’s account
        - Users to be able to report crashes with an attached bug report
        - To request access to user’s swipe and browsing information

- **UI-User interactions:**

  The articles will present the user with a few options: Slide-Right “Like”, Slide-Left “Dislike”, and Select a Heart to “Favorite”. Upon performing either of the two swiping actions, the user will experience a smooth transition to a new article. If the user “Favorites” an article, the user will place that article into a queue structured library that is viewable via private user options.

    - As a user, I want

        - To have a tutorial of the application’s features.
        - To change the appearance of the application. (if time allows)
        - To see the news articles organised in a short, and concise manner in my news feed.
        - To have a feature in which the selected articles are presented, in full, one at a time to the user.
        - To see visual feedback when I swipe on an article.
        - To have the option of autoplay on embedded videos in articles.

- **Database Management:**

  The database should perform fetching and storing operations according to user input delivered through the multi-layer structure of the application. Fetching articles from the database will consider user preferences to provide a completely customizable source of modern news. The database controls will store a user’s personal preferences as it applies to their current instance of the application.

    - As a user, I want

  - To see a range of articles related to the category I search so that there isn’t a lag based on some of my later swipes.
  - See an updated list of articles based on my swipe pattern which will filter based on left and right swipes.

- As a developer, I want

  - a database that updates regularly with recent news articles
  - a sorted database where I can retrieve articles related to a certain category
  - a database with articles tagged by keywords which allow easy searching for articles based on certain keywords.

Design Outline

The key design components are the application hosted on the client, a light server layer that will mask the database and manage user requests, and a database which will interact with the server layer and store news articles. The system is based on a direct Client interaction directly to a light server layer within the database server model integrated with components of the MVC architecture will include a complete front-end interaction which communicates user requests to a controller that interacts with the database.

- Client
    - The client will download our app which will load necessary program functionality for client to server interactions upon start-up.
    - The client will host application functions such as UI and behind the scenes requests and gets to and from the server.
    - The user will specifically be shown an initial account setup or login page, then forwarded to the most recent news feed with menu options to explore alternate features.
    - Backend work on the client side will consist of checks and requests pertaining to each respective UI feature.
    - Send requests to the database server to be interpreted by the light server layer.
- Light Server Layer
    - This will serve as a mask to interpret requests from the user for the database for the purpose of pulling and preparing information from the database and return it to the client.
    - Given information from the user regarding their channel topic, likes, and dislikes the server will interpret the data in preparation for a request to the database.
    - The server will finally pull proper information from the database and send the information to the client per request.
- Database

- The database will store information regarding news articles, a user list.
- The database will have a light layer to communicate requests from the client to pull correct data per user request.
- The database will store news articles, and be updated periodically to ensure relevant news is being presented to the user.
- The database will also store instances of users and newsfeeds to be communicated upon application startup.

Design Issues

- **Issue 1**

  Number of Articles Supplied to User

  - Option 1: Supply 1 article at a time to user from database, update like/dislike algorithm after each swipe on an article.
  - Option 2: Supply several articles at a time to user, update like/dislike algorithm after all articles are swiped.
  - Option 3: Supply several articles at a time to user, but have a buffer of new articles based on swiped interest when the user is almost done swiping.

  Decision:

  We chose to go with option 3 because of the time it will save searching for new articles. Option 1 would be too costly because it would ask for a new article after every swipe, which would involve searching the database, retrieving an article from the database and passing the article back to the user. Option 2 is a mediocre option, but there would be noticeable lag after the user swipes all the articles and the controller is searching for new articles that would interest the user. Option 3 allows the application to grab multiple articles at a time, and have several new articles lined up after the initial articles were grabbed to allow a seamless flow for the user. As a team, we decided this allowed for the fastest performance of our application with minimal wait time.

- **Issue 2**

  How should we obtain news from online sources?

  - Option 1: obtain news articles from available developer API's and parse the JSON files for necessary data.
  - Option 2: use a web crawler mixed with a data mining algorithm to retrieve the news from a plethora of news sources.

  Decision:

  It has been decided to use option 1, obtain news articles by utilizing developer API's for top online news sources. Most developer API's accept a search query and return all available news articles in the form of JSON. This JSON would then be used to update the database and return news articles for the user.

- **Issue 3**

  How do we apply user updates to fetches from the database?

- Option 1: update the database as the user makes swipes. This would be very time inefficient.
- Option 2: load the database based on the categorical choice of the user initially. Although this would take a bit more time initially, it would save the user time later on in terms of changing keywords based on swipes.

Decision:

We chose to go with option 2 because it would save more time and is more efficient than option 1. Option 1 would be costly because even though we give the user articles exactly based on their category and keywords right at the beginning. This would involve updating the database and providing the user with new articles as they changed their choice and update the database constantly along with this.

- **Issue 4**

How would we integrate the update of database and categories based on user needs and rest of the Class implementations?

- Option 1: Update user information in the database as updates occur
- Option 2: Overwrite user information when the application is shutting down

Decision:

The database will be updated using option 2. Upon startup, user information will be stored locally on the user's device. During use, updates will take place locally. Upon shutting down, local updates will be sent to the server, where the database will be updated.

- **Issue 5**

Categorizing the news

- Option 1: Present the user articles from all news categories
- Option 2: Allow the user to select a category of news and filter news based on the specific category.

Decision:

As a team, we decided filtering the news by categories was the best decision. Users might have no interest in a category, so as developers we should allow the user to avoid the hassle of swiping articles they had no initial interest in.

- **Issue 6**

Selecting a client-server architecture

- ○ Option 1: Design the architecture using SQLite.
- ○ Option 2: Design the architecture using MySql, PHP, and Apache
- ○ Option 3: Utilize SQLite for to manage local data, while utilizing MySql for user data

Decision:

Newslog's client-server communication will be designed using SQLite for local data and WAMP (Windows, Apache, MySql, PHP) architecture for user data. Upon startup, a user will be prompted to log in. The user's information will be requested using MySql, and will then be saved using SQLite. Then the user's information will be saved locally, until the application closes. Then, MySql will update the database on the server.

- **Issue 7**

Likes/dislikes in keywords

- ○ Option 1: Incorporate dislikes into keyword algorithm
- ○ Option 2: Only consider likes in keyword formation
- ○ Option 3: Incorporate a weighted point scale for keywords based on likes and dislikes

Decision:

Option 3 was selected because it provides the most accurate way to gauge user interest. Liking a certain keyword multiple times must mean that the user displays a strong interest on the topic. Therefore, the system would keep track of the number of times the user liked a certain keyword to display the appropriate articles.

- **Issue 8:**

How often do we update the database for new news?

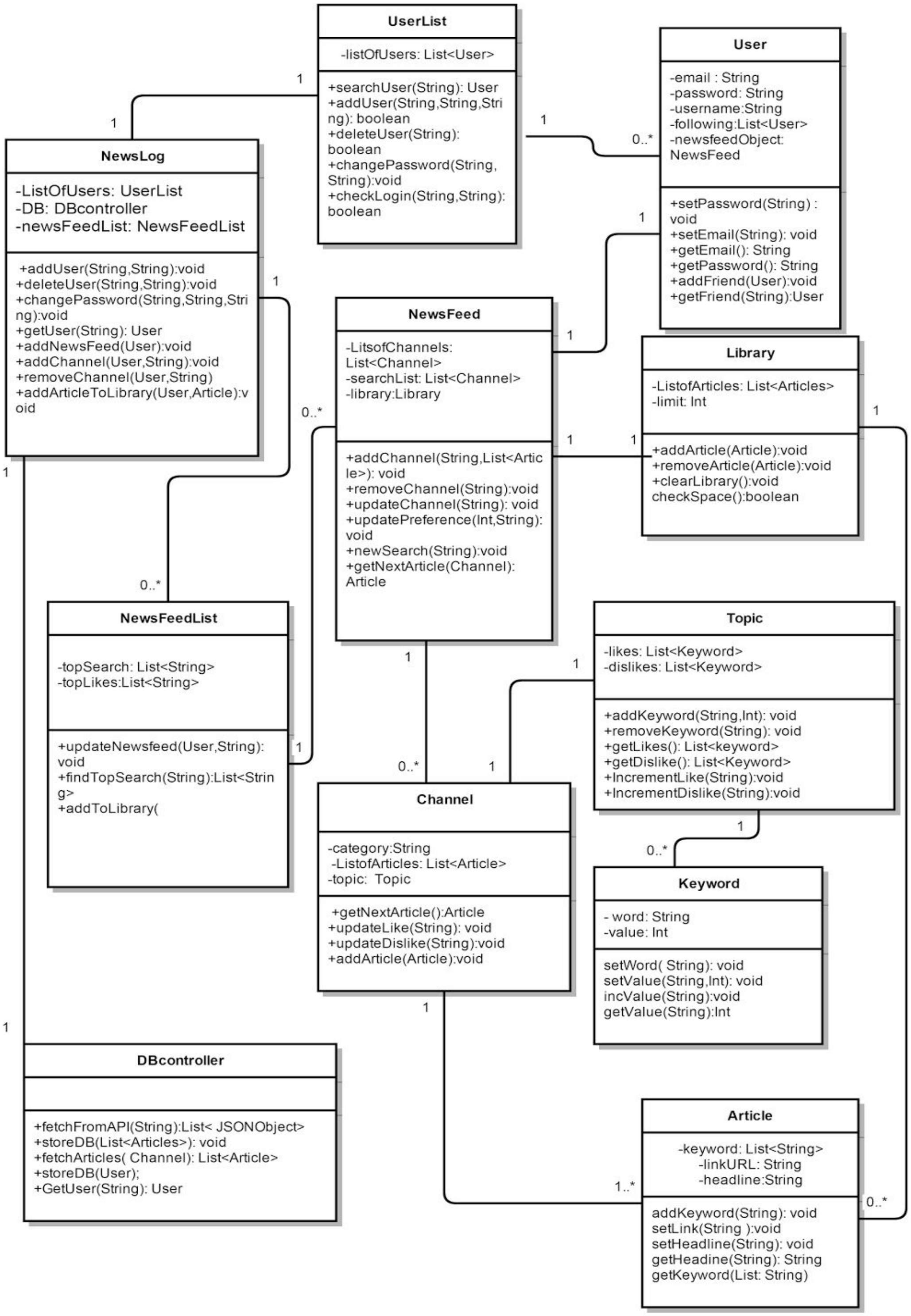
- ○ Option 1:Incorporates an update once every week.
- ○ Option 2: Incorporates an update once everyday.
- ○ Option 3: does an update every hour

Decision:

Option 3 was selected because it provides an accurate timeframe in which we need to update the database. By updating the database approximately once every hour we get enough news updates for that hour. Any less than that and frequency of an update wouldn't be enough.

Design Details

Figure 2.1: Domain Classes


UserList
-listOfUsers: List<User>
+searchUser(String): User
+addUser(String,String,String): boolean
+deleteUser(String): boolean
+changePassword(String, String):void
+checkLogin(String,String): boolean
User
-email : String
-password: String
-username:String
-following:List<User>
-newsfeedObject: NewsFeed
+setPassword(String) : void
+setEmail(String): void
+getEmail(): String
+getPassword(): String
+addFriend(User):void
+getFriend(String):User
NewsLog
-ListOfUsers: UserList
-DB: DBcontroller
-newsFeedList: NewsFeedList
+addUser(String,String):void
+deleteUser(String,String):void
+changePassword(String,String,String):void
+getUser(String): User
+addNewsFeed(User):void
+addChannel(User,String):void
+removeChannel(User,String)
+addArticleToLibrary(User,Article):void
NewsFeed
-LitsofChannels: List<Channel>
-searchList: List<Channel>
-library:Library
+addChannel(String,List<Article>): void
+removeChannel(String):void
+updateChannel(String): void
+updatePreference(Int,String): void
+newSearch(String):void
+getNextArticle(Channel): Article
Library
-ListofArticles: List<Articles>
-limit: Int
+addArticle(Article):void
+removeArticle(Article):void
+clearLibrary():void
checkSpace():boolean
NewsFeedList
-topSearch: List<String>
-topLikes:List<String>
+updateNewsfeed(User,String): void
+findTopSearch(String):List<String>
+addToLibrary(
Topic
-likes: List<Keyword>
-dislikes: List<Keyword>
+addKeyword(String,Int): void
+removeKeyword(String): void
+getLikes(): List<keyword>
+getDislike(): List<Keyword>
+IncrementLike(String):void
+IncrementDislike(String):void
Channel
-category:String
-ListofArticles: List<Article>
-topic: Topic
+getNextArticle():Article
+updateLike(String): void
+updateDislike(String):void
+addArticle(Article):void
Keyword
- word: String
-value: Int
setWord(String): void
setValue(String,Int): void
incValue(String):void
getValue(String):Int
DBcontroller
+fetchFromAPI(String):List< JSONObject>
+storeDB(List<Articles>): void
+fetchArticles(Channel): List<Article>
+storeDB(User);
+GetUser(String): User
Article
-keyword: List<String>
-linkURL: String
-headline:String
addKeyword(String): void
setLink(String):void
setHeadline(String): void
getHeadine(String): String
getKeyword(List: String)
1
1
1
0..*
1
1
1
0..*
1
1
1
0..*
1
1
1
1
0..*
1
1
0..*
1
1
1..*
0..*

User
A User represents all information stored about a user in the system including the email,username,password etc.

UserList
This class manages the users in the system and handles different methods like adding new users, updating passwords, verifying email password pairs etc.

Article
An Article represents a news articles that will be fetched from different news sources using APIs.It will contain different information about the article such as heading,keywords,time stamp etc.

Library
Library class stores all the articles that the user saves to read later. The library has methods of adding new articles and deleting articles.Library keeps a limit on the number of articles stored.

NewsFeed
The Newsfeed represents the contents that the user can request to see. It contains different channels (added according user’s preference).The channels will be updated for more articles automatically and on request. Newsfeed contains the library that the user uses to store articles. Newsfeed also manages the search queries made by the user.

NewsFeedList
This class manages the overall working of newsfeed for each specific user. It will implement different methods to analyze the news activity of various users. This information will help in determining the ongoing trends,most searched keyword etc. The algorithm will make use of this information for fetching new articles and populating each user’s feed according to his/her preferences.

Channel
Channel represents the stream of articles on a specific category. It will implement methods to keep a track on user’s reading activity including likes and dislikes.

Keyword
This class represents the word and like/dislike counter pair.

Topic
This class maintains a list of most liked and disliked keywords for each specific category.

DBController

Interactions between the database and the rest of the system will be done through this class. DBController will implement different methods like fetching new Articles from the internet and storing them on the database.

NewsLog
Newslog represents the main controller class. It satisfies the requests received from the client by delegating work among the sub-controller classes.

Figure 2.2: Sequence Diagram Depicting User's Request for an Article
The following diagram depicts a typical sequence of events when a user requests a category specific article for the first time. If the news controller already has enough articles loaded into the list of articles that it stores, then the news controller will immediately satisfy the user's request for a new article as opposed to requesting a list of articles from the database controller.

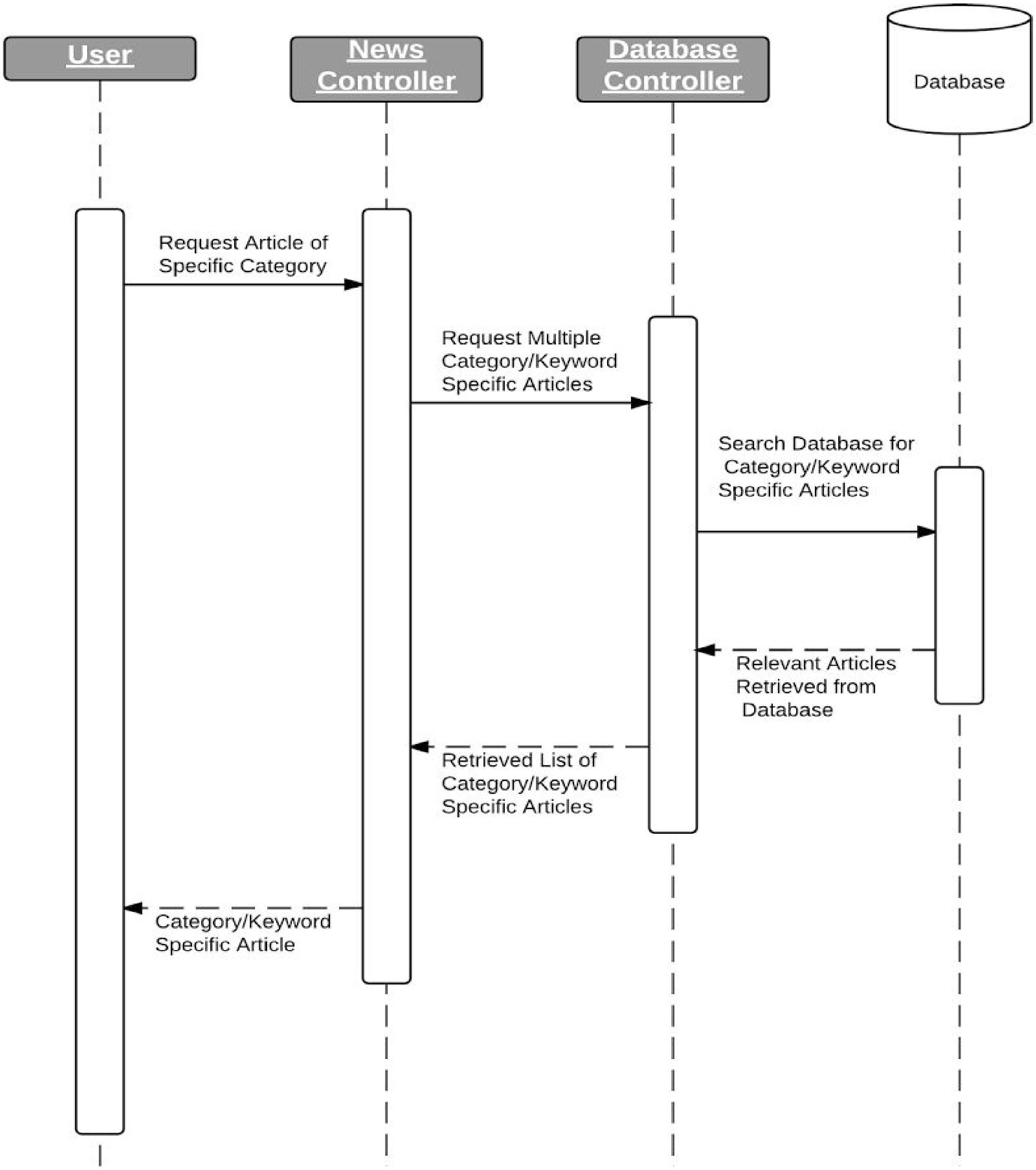

User
News Controller
Database Controller
Database
Request Article of Specific Category
Request Multiple Category/Keyword Specific Articles
Search Database for Category/Keyword Specific Articles
Relevant Articles Retrieved from Database
Retrieved List of Category/Keyword Specific Articles
Category/Keyword Specific Article

Figure 2.3: User Request to Add Article to Saved Library

The following diagram depicts the typical sequence of events that occurs as a result of the user adding an article to their saved article library.

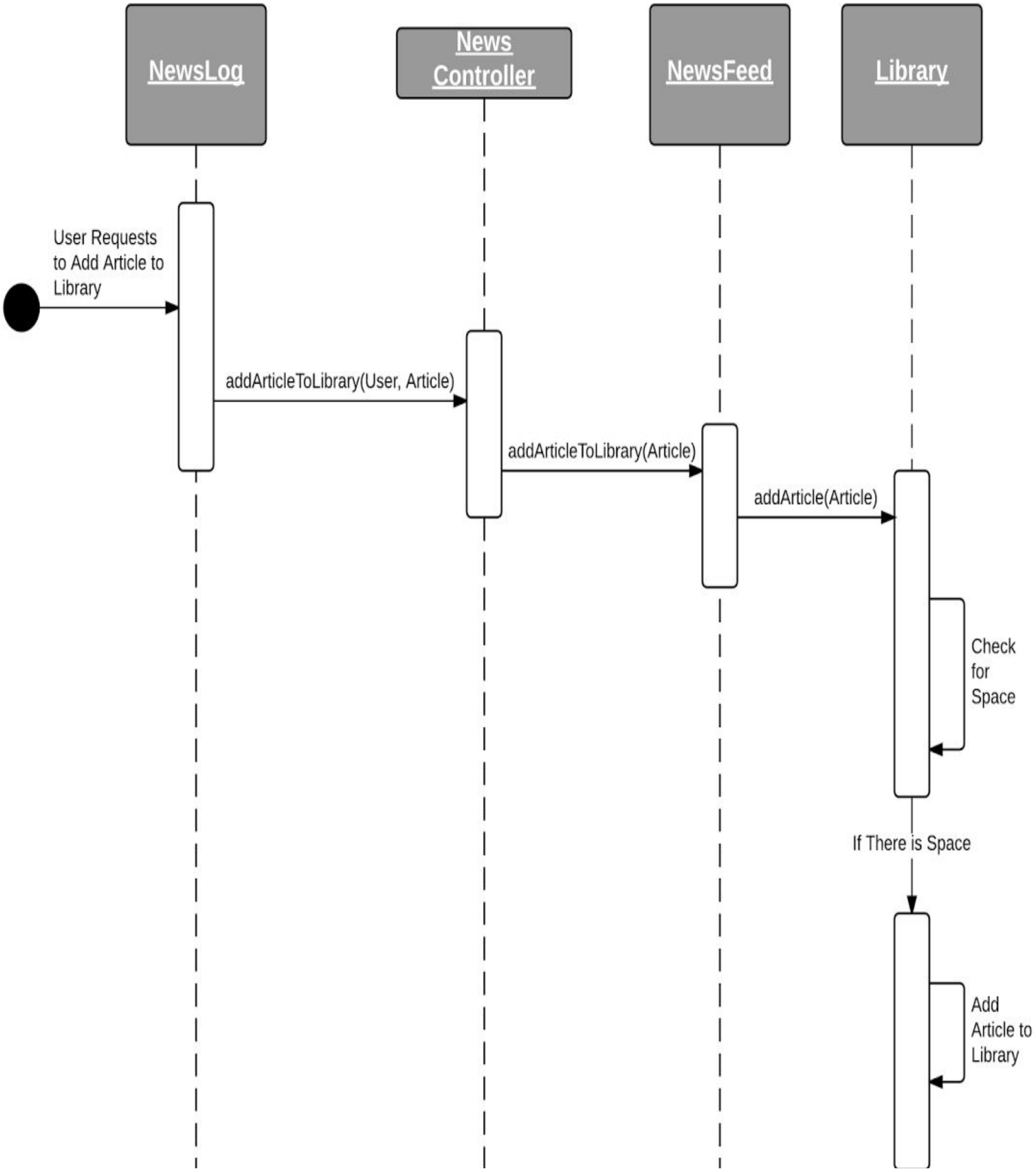


Figure 2.4: Fetching Articles from the News Source

The following diagram depicts the polling that the Database Controller is constantly performing at set intervals in order to fetch articles from the news sources.

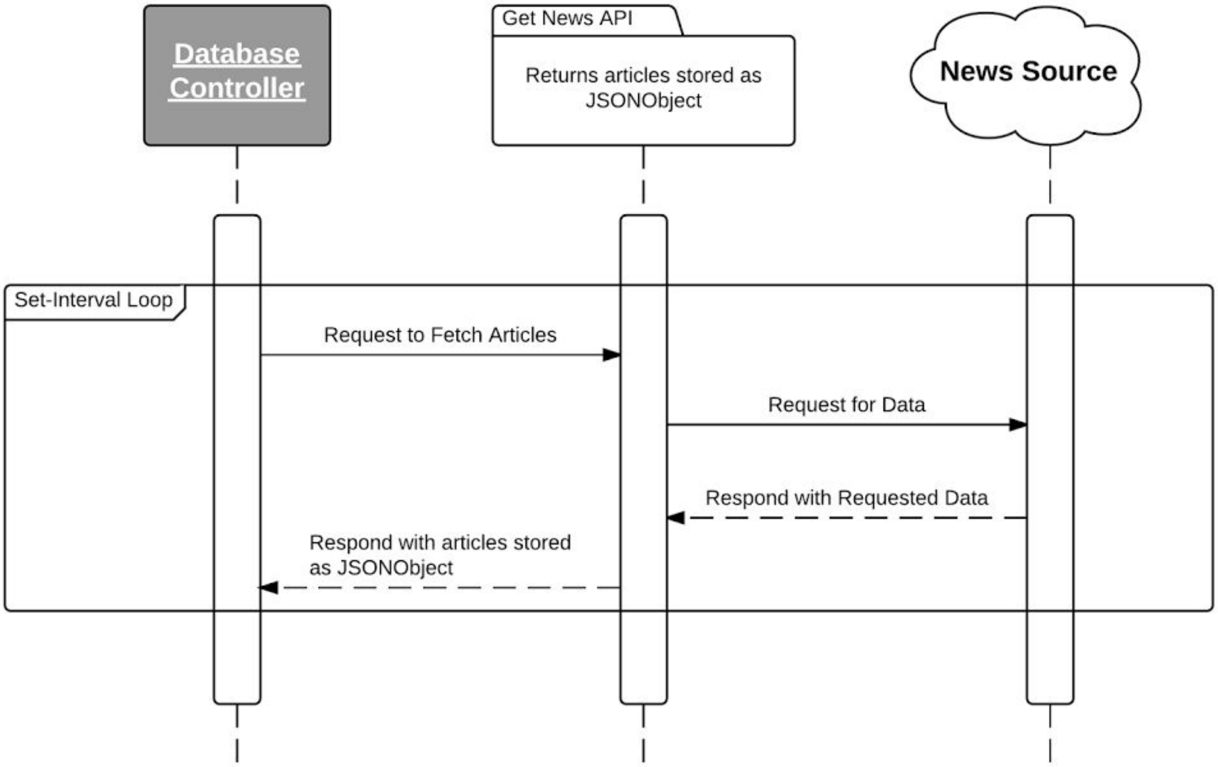

Database Controller
Get News API
Returns articles stored as JSONObject
News Source
Set-Interval Loop
Request to Fetch Articles
Request for Data
Respond with Requested Data
Respond with articles stored as JSONObject

Figure 2.5: User Account Creation Activity Diagram

The following action diagram displays how a user can sign up for an account. On the left side is the tasks the user will be presented with and on the right side is the server backend making sure all the credentials entered are valid.

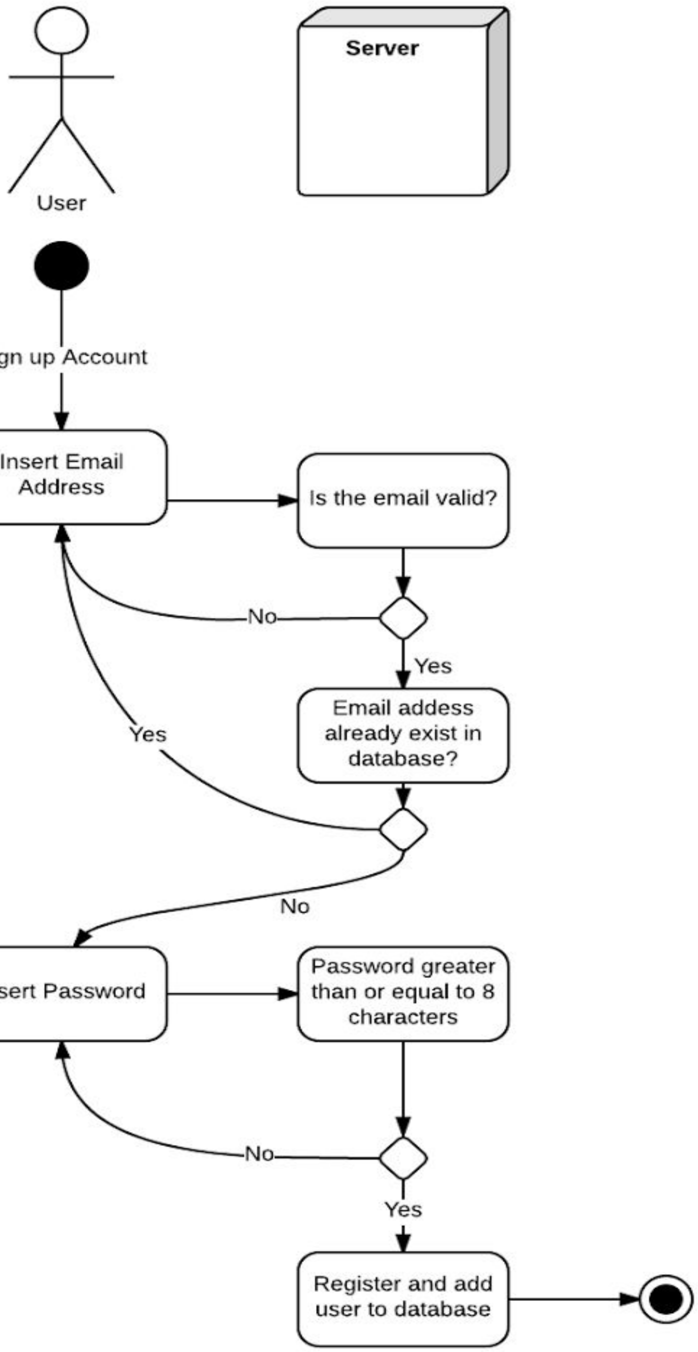
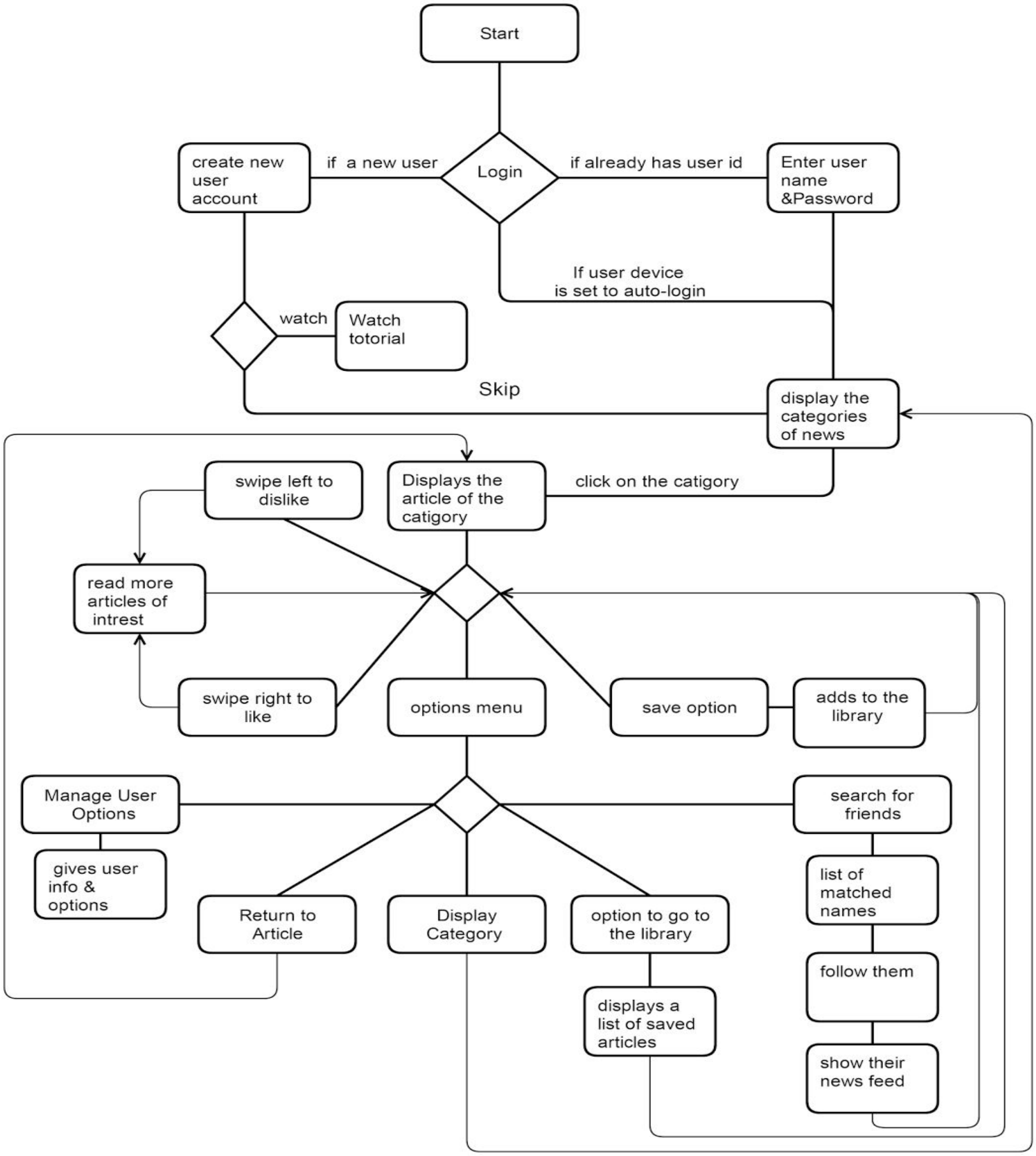

User
Server
Sign up Account
Insert Email Address
Is the email valid?
No
Yes
Email addess already exist in database?
Yes
No
Insert Password
Password greater than or equal to 8 characters
No
Yes
Register and add user to database

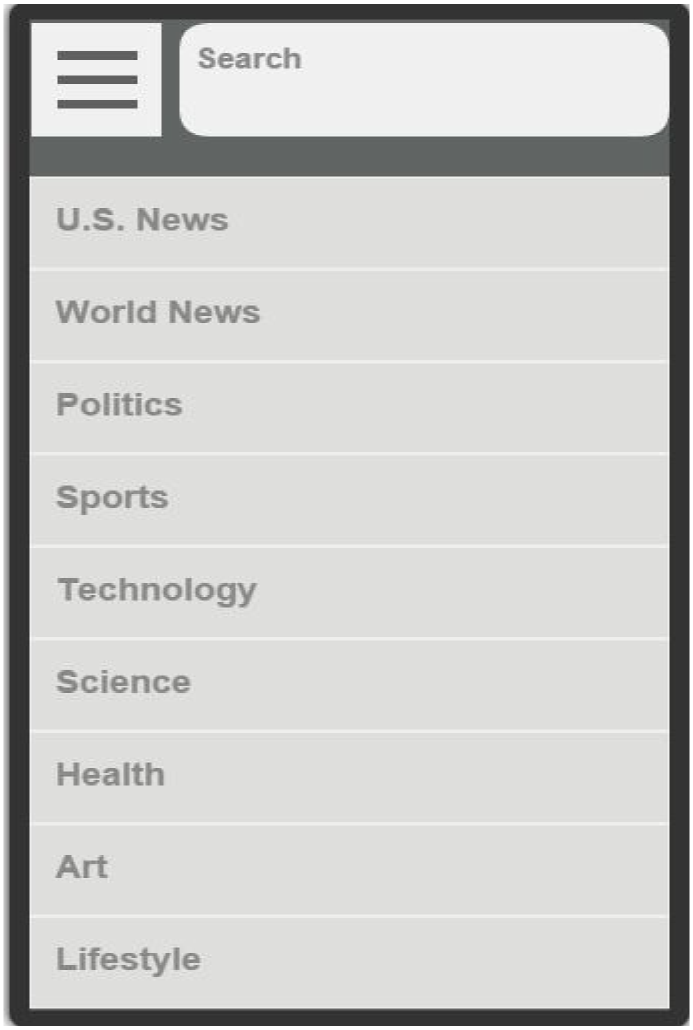
Figure 2.6: Activity Diagram

The Activity Diagram presents a general overview for the application options, and connections that will be built into the UI. We included decisions available to the user and application responses according to user input. The main functionality of our application to continually relay news to the user will be made possible by the constant return to the article display and once again waiting for the user's next decision.



GUI Mockup

List of categories to choose from



Search

U.S. News

World News

Politics

Sports

Technology

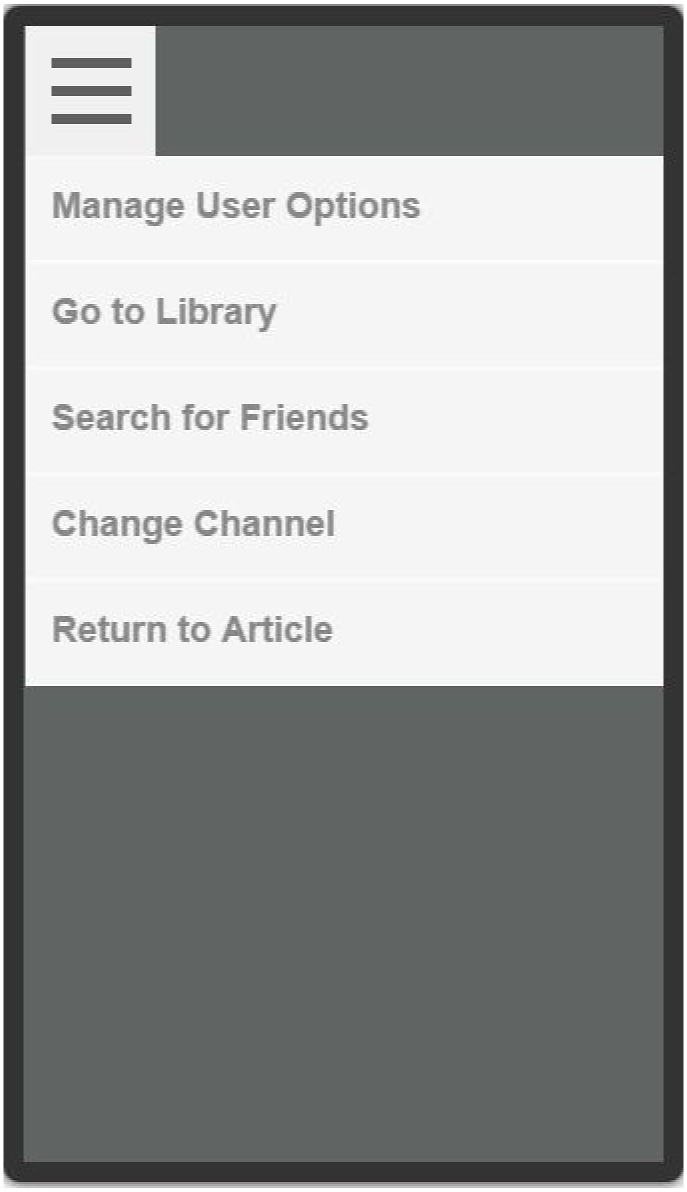
Science

Health

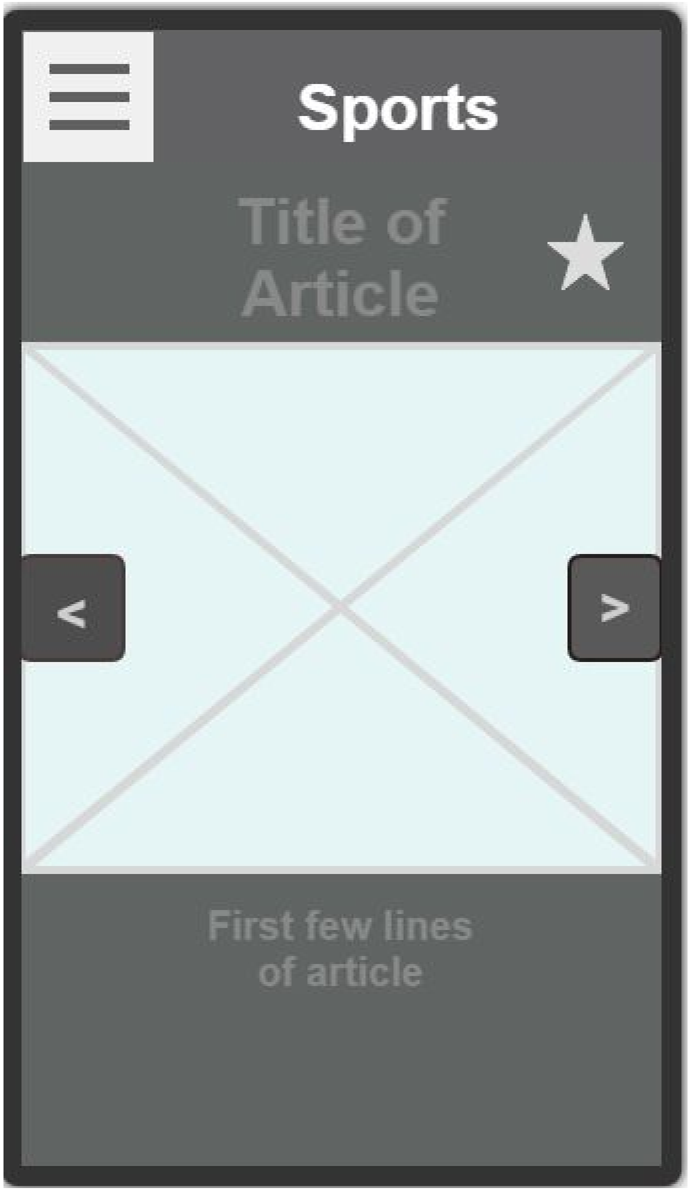
Art

Lifestyle

List of Menu Options



News Presentation


Sports
Title of Article
<
>
First few lines of article