Muhammad Asmar, Jordan Barrios, Kyle Costela, Parker Hawkins

Group 08

Professor Huang

CEN 4010 Principles of Software Engineering, Summer 2020

14 July 2020

Milestone 3 — COVID-19 Website

Executive Summary

The final product of this project will be an informative news-website focused on the ongoing COVID-19 pandemic. The website will use two main APIs. One will be a coronavirus API for raw data, the other will be a news API which will filter for articles mentioning the pandemic across multiple sources. Each article will have a related discussion forum in which users can comment their thoughts on the article. Due to normal news sources having a focus in a variety of areas, it can sometimes be hard to find up-to-date information on the pandemic specifically. The main advantage of this website is it will compile all these articles into one place. Additionally, the private forum will allow users to express their thoughts and feel connected.

Competitive analysis

This section illustrates the main differences between this website and other similar news sites. For the purpose of keeping people informed constantly on the state of the pandemic, this website should prove to be better than its competitors. These are major features that are planned to be implemented in the final product.

Key Features	Our Product	FOX	CNN
Pandemic-Targeted Discussions	✓	X	X
Account Creation	✓	✓	✓
Realtime Pandemic Data	✓	✓	X
Multi-Browser Compatibility	✓	✓	✓
Email-Directed Notifications	✓	✓	✓

Our plan is to create a more specialized and consumer-oriented product that is specific to the COVID pandemic. Our product will allow customers to see data in realtime about the pandemic and allow them to participate in discussion forums related to various issues around the pandemic. Some of the planned advantages pertaining to our site is how specific it will be to the COVID pandemic. It is hard to find news outlets that are extremely specific to one area let alone a virus and that is a major advantage for our product. Another key feature that gives us an edge will be our discussion forums. It is hard to communicate with people online in forums similar to websites like FOX and CNN, however, due to how specific our platform will be, it allows for easier and more direct communication.

Data definition

Site/Website - A general use term to refer to the end product as a whole.

User - Any person who uses the product, doesn't necessarily need an account.

Commenter - Any user who uses the dedicated space below a News article for discussion or comments in the respective comment section.

Account - Any account owned by an end user.

Administrator/Admin - Any person involved in the end product with either managing comments, forum posts or general maintenance on the site.

Data Source - Refers to the API for statistical data for Covid-19 cases around the world.

News API - Refers to the API for news articles regarding anything Covid-19, could be opinion pieces, predictions, background information, contains more than just statistical data.

News Source - Refers to the specific publisher of a news article, i.e. CNN, FOX, Jalopnik, e.t.c.

News Article - Refers to a specific video, audio clip, or written piece brought forward by the News API.

Comment Section - The space allotted on every news article page for discussion of the related media. -limited to 500 characters.

Forum - A more general use forum then the Comment Section, for discussion. Allows users to comment or generate posts regarding Covid-19 as a whole instead of having to be related to a particular News Article. Also Allows Administrators to communicate with the users with relevant information regarding the website or otherwise.

Overview, scenarios and use cases

Overview:

This website makes use of two main APIs. The first is the NewsApi from newsapi.org. This API is able to search by author, date, title, and content. By making use of this API,

this website will be able to compile all the latest news on the ongoing pandemic. It can also provide the related images, with which this website can create custom thumbnails for each article.

The second API this website will use is the coronavirus data API from corona.lmao.ninja. This API provides up-to-date information and statistics on the ongoing pandemic, updated every 10 minutes. Not only does it provide numbers of infected and deaths, but also on numbers on recovery and tests administered. This information is important in monitoring the state of the pandemic. Users can search by country. By combining these technologies, this website will be a 24/7 coronavirus-dedicated news source.

Additionally, there will be a comment section beneath each article, where readers can express their thoughts and feelings about the article and the situation in general. Each comment has time-posted, the username of the poster, and their profile image. The comment section allows people to make original comments or reply to someone else's post. This is aimed to provide a sense of community, where other news sites may feel uninviting.

Scenarios:

User creates an account:

-Initial Assumption: A user has decided they would like to make an account in order to comment on articles. They are on the front-page of the website.

-Normal: Starting from the front-page, the user clicks on the "sign-up" button. The user is then prompted to make a username and password. Users are prompted to enter the password twice, to ensure they match. On submission, the website confirms the creation of the account and brings the user back to the front-page.

-What can go wrong: In the case where the passwords do not match, the user will be prompted to type both passwords again.

-Other activities:

-System state on completion: When brought back to the front-page, the user is automatically logged in. The user is free to browse and comment on the website.

User browses website:

-Initial Assumption: A user has decided they would like to see up-to-date news regarding the pandemic. They are on the front-page of the website.

-Normal: The front-page of the website shows some articles. By scrolling down, the user finds more articles, ordered by most recent. The user finds an article they are interested in and clicks on it to view more. Once done, they can click at the top of the page to return to the front-page.

-What can go wrong: If for some reason the article does not load when clicked on, the website will inform the user and bring them back to the front-page.

-Other activities:

-System state on completion: When brought back to the front-page, the user can continue scrolling as before.

User makes a comment:

-Initial Assumption: After browsing and reading an article, the user has decided they would like to make a comment. They are on the article page.

-Normal: Starting from the article page, the user scrolls down to find the comment section. The user clicks on the text field and begins typing their comment. Once done, the user submits the comment.

-What can go wrong: If for some reason the comment fails to post on submission, the user will be informed and asked to try again.

-Other activities:

-System state on completion: On successful submission, the user is informed the comment was posted and is brought back to the article page.

Use Cases (stories):

-Seeking Up-to-date Information:

A business-owner, Jill, is seeking up-to-date information on pandemic so she can plan her next move accordingly. She is only interested in news regarding the pandemic, but many

news sites include many articles that are irrelevant to her interest. After searching for a bit, she stumbles across this website. She sees that the website not only has constantly updated statistics regarding the pandemic, but also has related news articles from a variety of sources. Additionally, she finds it interesting that there are discussion posts separate from the articles, where people share their thoughts on the situation. Jill decides to make an account so she can comment on the articles as well.

High-level function requirements

1. Discussion Board/Comment Section(**Priority 1**): Beneath each article, there will be a comment section related to the article. If the user is logged in, they have the option to make a new comment or reply to someone else's. The comments are ordered from most to least recent.

1.1 Usernames and profile pictures, if uploaded, are attached to each respective comment.

2. Account Creation(**Priority 1**): The user has the option to create an account in order to make comments on articles. Username is visible to others when they make a comment.

2.1: Once user has entered their registration information, the page will validate to make sure that there are no other users in the database with the same data.

2.2: The user shall not be allowed to participate in any of the websites features without having a valid account created.

3. Pandemic data(**Priority 1**): Appears either at the top of the page or on the sides. It is always visible regardless of how far the user scrolls.

3.1: The data will be visible at all times on the screen. However, the user will have the option to minimize the displayed pandemic data to enhance their browsing experience if they desire.

4. Email Notifications(**Priority 2**): Separate page that is available once the user has created an account. It will allow the user to participate in exclusive weekly notifications that are sent to their email.

4.1: This feature will require the user to simply enter their email and then hit a submit button. The email will be added to a separate table in the DB and validation will happen in the backend.

4.2: This page will only be displayed to the user if they are logged into their account.

Non-functional requirements

Security: the coronavirus website will allow the user to create a username and password before being able to use our service. Upon creating an account, the password will then be encrypted before it is stored on a database. When the user logs in, the username and password will be compared to the database if it exists before allowing the user to enter the website. Security can be quantified by the percentage of users that do not have to change their username or password after account creation.

Usability: the website will straight-forward to a person to use. When the website opens, the user will be presented with a login and create account form. They will enter in information for the corresponding form they wish to fill out. When successfully logged in, the user will be taken to a main html page that will have a form with only one entry. In the text input form, the user will give a country name. When the submit button is clicked, coronavirus information for that country will be displayed. Underneath the coronavirus data, there will be recent news with hyperlinks for the user to read more on coronavirus news.

Manageability: The login portion of the website will be managed solely by phpMyAdmin to store username and password information. To retrieve coronavirus and recent news information the API's listed below will be used to request information.

Portability: In order to ensure that our software is portable, our website must work across multiple web browsers including: Google Chrome, Internet Explorer, Mozilla Firefox, Microsoft Edge, and Safari on desktop and mobile devices. Using Bootstrap to develop our website will help because the templates are created to prioritize mobile devices. In

order to ensure our service works across all major web browsers, it will be imperative that we adhere to all HTML5 standards and not use any deprecated practices in web development.

Modifiability: With each modification of our website, the LAMP server will need to be updated with the changed files so that our website will be accessed with the latest changes and features. Also, each version of our website will need to be uploaded to github so that the changes can have pull requests to each developer on the team.

Performance: the performance of our website will depend mostly on response time to retrieve JSON data, parse it, and display it directly in an HTML file. Another performance aspect that can be measured will be the time it takes to create an account or validate login credentials by collecting user form data and inserting or comparing the information with username and passwords that are currently on the database.

Platform constraints: the account creating and login functionality will depend if there is a limit on the amount of usernames and passwords that can be stored in the accounts table on phpMyAdmin. There should be no platform constraints of any web browser that the user uses when on the website or if the user is using a desktop or a mobile device. Another platform constraint will be the use of the recent news API in our website. This is because to use the recent news API, the API only lets requests be made if the articles being requested are at most one month old.

High-level System Architecture

Client-Server Architecture

News API (GNews):

https://gnews.io/docs/v3?javascript#introduction

Coronavirus Data API:

https://corona.lmao.ninja/v2/countries/usa

High-level Architecture:

Client-Server Architecture

DB Organization:

Account table

username	**password**	**email**	**notify**	**img**
JordanB	mypassword12	mymail@gmail.com	true	profile.jpg
...	...	...		
...	...	...		

Article Comment Table

id	article_title	username	text
2	Pandemic Updates	JordanB	“This was a good article!”
...	...	...	...
...	...	...	...

Articles Table

article_title	comment_id []	img_url	main_text	article_url
Pandemic Updates	[2, 4, 7 …]	www.img.com/img.jpg	“Lorem ipsum dolor sit amet, consectetur adipiscing elit…”	www.anewsite.com/thisarticle
...	...	...	...	...
...	...	...	...	...

Media Storage:

Images for articles are provided by the newsapi.org api..

Images for users are stored in the database.

Search/Filter Architecture and Implementation:

For the article search bar, it searches by article title stored in the database.

For the coronavirus data search bar, its searches by country name. This is provided by the

api, not in the database.

For the login screen, it searches by username then compares the typed password to the stored password.

Our own API's:

none

Non-trivial algorithms or processes: If the user would like to create an account, they can add a username and password to the create account form. When the user clicks submit, a post method will be called to createaccount.php. From there, the password will be encrypted and together with the username, it will be added to phpMyAdmin relational database. On the database, there will be a users table created where there will be two data types: a varchar for the username and a varchar for the password. The login credentials will only be added to the database if there is not currently a username and corresponding password that is already stored in the database. If there is an error, then an error will be displayed. If the login credential is stored successfully, the user will be escorted directly into the coronavirus search.

If the user would like to login, they can enter in their login credentials into the login from. When they click submit, a post method call will be made to a login.php file that will retrieve the username and password from the database. When selecting data from the users table, the number of rows in the selected rows will be calculated to make sure that they equal one; if not, then an error will be displayed that either the username and

password does not match any stored on the database.

In the coronavirus search, there will be one input field to put in a country name. When the user enters a country name, an http request will be made to the coronavirus API listed in this document. There will be function in a javascript file that will create an http connection, parse the json data, and display the data in the document object model.

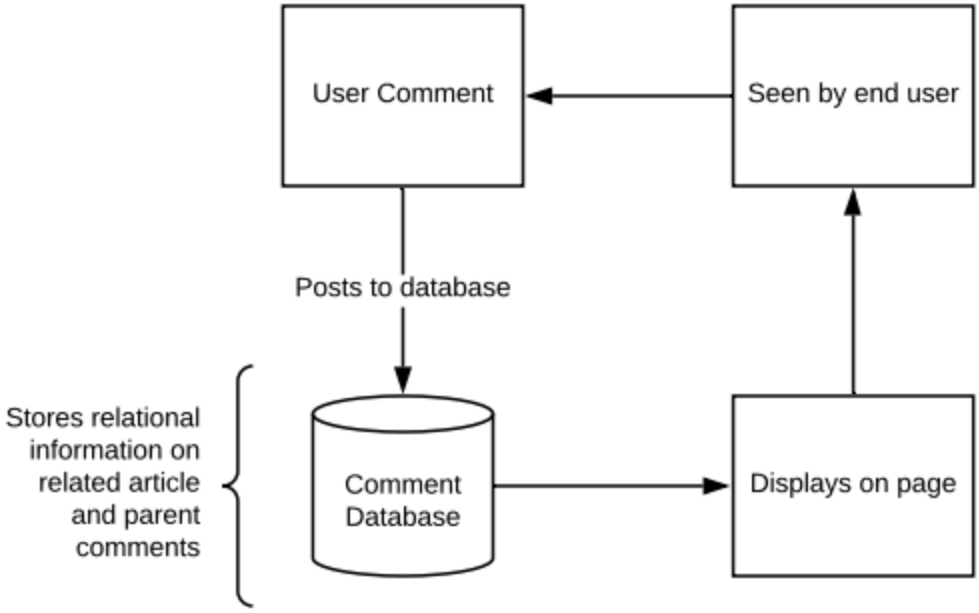
Below, will be a button to click on that will take the user to relevant articles related to coronavirus information with the ability to create comments and interact with other users of the website. There will be a separate table in the database that will hold the comments that users post and use that data to display a message directly into the document object model.

Finally, there will be a link in the navigation bar that will allow the user to sign out of their account. When clicked, the session variable that holds their account information will be deleted and the user will be escorted back to the login page.

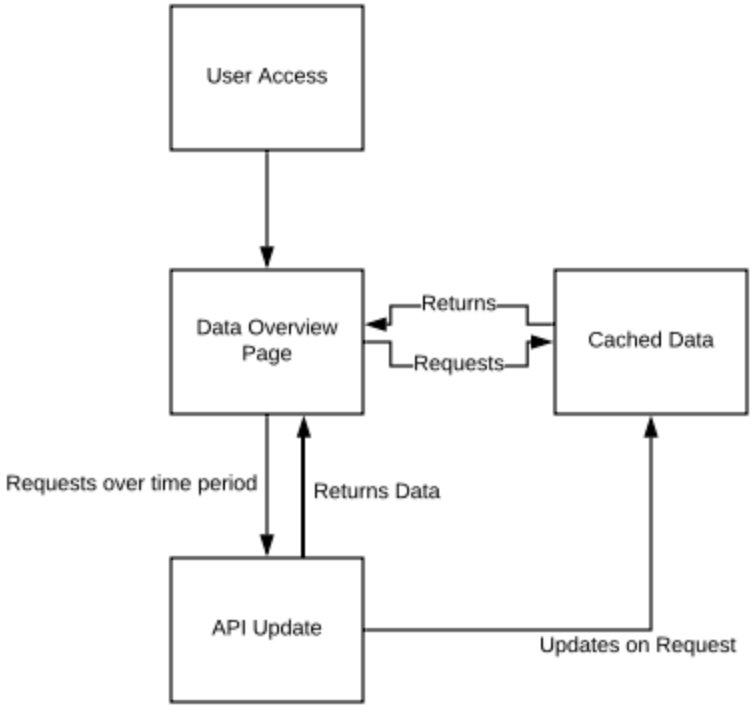
High-Level UML Diagrams

Class Diagrams:

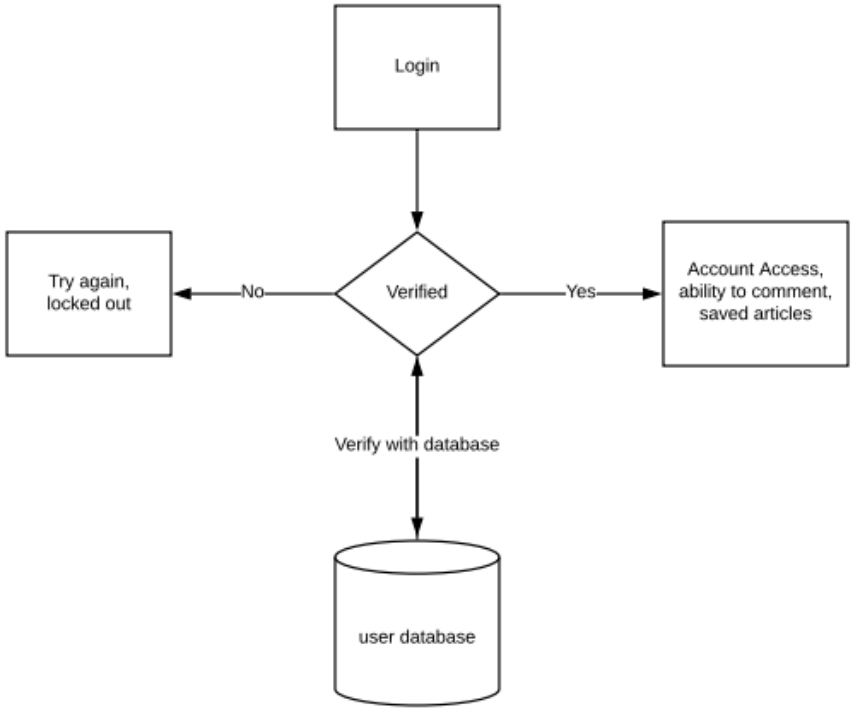
Comment UML



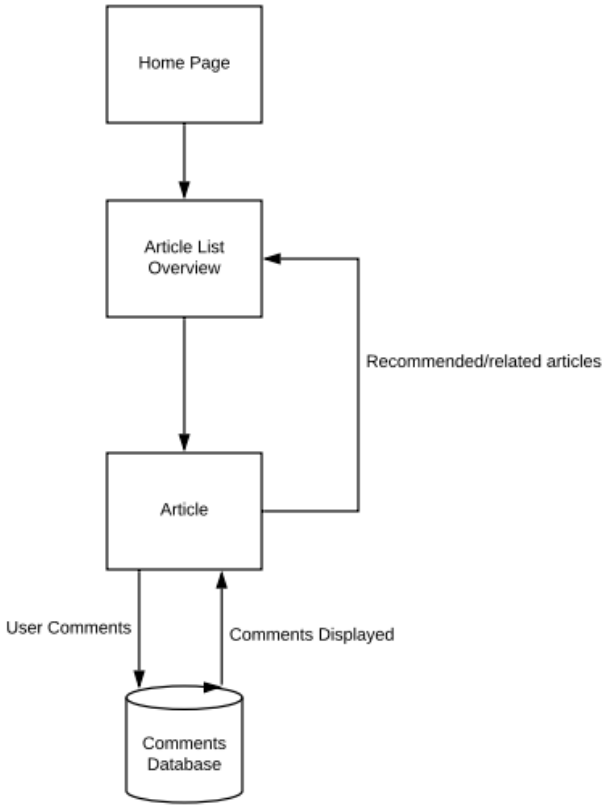
Data API Overview UML



Login Overview UML

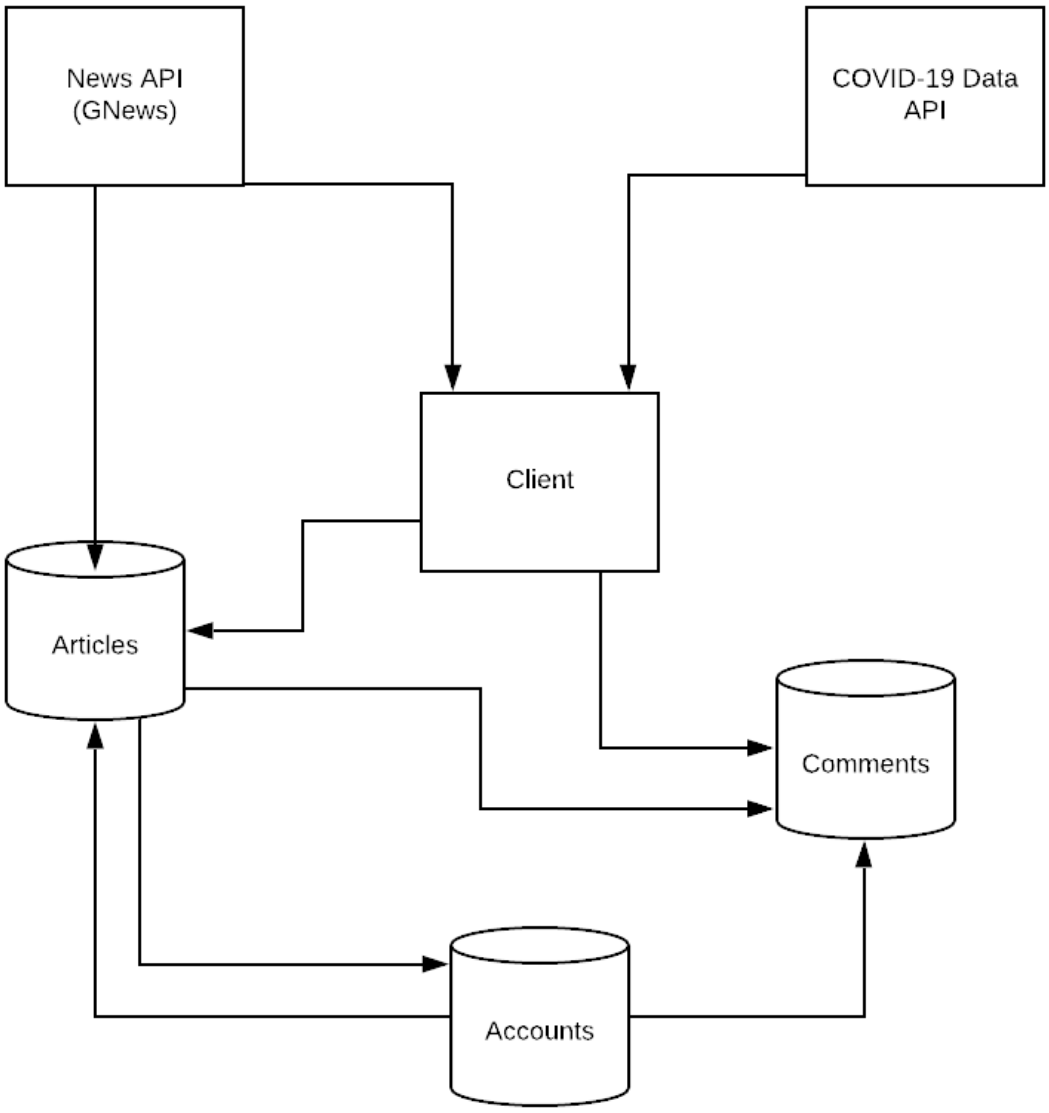


Article Overview UML



Component and Deployment Diagrams:

High-Level Component Diagram



Key Risks

Skills Risks:

Few of us have extensive experience with building fully functional websites. However we all have some basic experiences with the various tools individually, from previous courses. Already we are helping each by doing research on the various tools and features of the website. We have and will continue to show each other online resources and tutorials on how to build each feature.

Schedule Risks:
Because we do not have much experience, everything will be slow to start in terms of implementation. Even the tutorials themselves will take up precious development time. As such, we will plan to start development early. This will provide us with the biggest time cushion possible.

Technical Risks:
We were having some trouble with the initial news api that we had proposed in Milestone 1. Specifically, difficulty arose in getting proper authentication with a free developer account. Already, we have decided to drop the original api for the GNews api. Although not as robust as the previous one, it will still provide sufficient data and information for our purposes.
2. One issue that may arise with the development of our website will be the utilization of the API's that we have selected. One of the news API's that we had selected has been terminated from being free to use and now has a premium pay for service. Another issue related to the API's may be if the API has its service terminated.

Team

Scrum Master: Kyle Costela

Product Owner: Jordan Barrios

Project/Development Team: Muhammad Asmar, Parker Hawkins

Checklist

a) Team decided on basic means of communications: DONE

b) Team found a time slot to meet outside of the class: DONE

c) Front and back end team leads chosen: DONE

d) Github master chosen: DONE

e) Team ready and able to use the chosen back and front-end frameworks: DONE

f) Skills of each team member defined and known to all: DONE

g) Team lead ensured that all team members read the final M3 and agree/understand it before submission: DONE