

Williamston Website Project

A Project Report
Presented to
CMPE-131
Fall, 2023

By

Erickson Ng, Hannah Tran, Jason Huynh, Viet Nguyen

12/08/23

Copyright © 2023
Erickson Ng, Hannah Tran, Jason Huynh, Viet Nguyen
ALL RIGHTS RESERVED

ABSTRACT

News Posting

By Erickson Ng, Hannah Tran, Jason Huynh, Viet Ngyuen

The proliferation of information in today's fast-paced world has solidified the importance of online platforms in delivering timely and relevant news and information to users. The city of Williamston, Michigan initiated this project to improve the user experience of the website's news sections. The current news section on the website lacks structure and a vast selection, making it difficult for users to find content relevant to their interests. The primary objective of this project is to create a user-friendly interface that offers a diverse selection of news articles categorized into local, state, and national news sections for increased convenience and relevance. On the homepage of the website, visitors will be able to see local news and public announcements for the city of Williamston that are written by city administrators. The content written for local news will be dated and archived after a certain duration like a month after posted.

For state and national news, an API that curates news (called NewsAPI) will be implemented so that users have access to recent news content from multiple news outlets. Users will also have access to a search bar integrated with the nav bar which will display corresponding articles based on the search. An RSS Feed will be implemented to display the latest posts pertaining to different categories such as health, business, technology, and sports. Users will be able to choose which categories they want to see on a dedicated feed page. As an enhancement to the website, a responsive UI will be implemented to support various devices. Additionally, admins will be able to create local city posts, edit them, and edit public announcements. As part of improving the UI, the outsourced state and national news will be displayed as large cards with headlines and a brief summary. The cards will be clickable and will lead to the news outlet that wrote the piece.

The IDE being used will be Visual Studio Code, as it is a common IDE for coding in React, which is the Javascript library that will be used for creating the user interface for the news contents. In conjunction with React, a React components library called Material UI to simplify the creation of UI components. The use of Material UI also ensures that the website components have responsive sizing. Firebase Authentication will be used to handle user login and signup services while Firebase Realtime Database will be used to create Williamston posts, edit existing posts, and handle city announcements.

Acknowledgements

Gratitude goes out to Dr. Andrew Bond for his guidance in this project's development in aspects of the deliverables for the website and project report. Appreciation is also in store for classmates of CMPE 131 Section 4 at San Jose State University for giving feedback on improvements to project documentation and for observing the project update presentations on the dates of 10/9/23 and 12/04/23.

Table of Contents

Chapter 1 Introduction

1.1	Project goals and objectives	Page 01
1.2	Problem and motivation	Page 01
1.3	Project application and impact	Page 01
1.4	Project results and deliverables	Page 01
1.5	Market research	Page 02
1.6	Project report structure	Page 02

Chapter 2 Project Background and Related Work

2.1	Background and used technologies	Page 03
2.2	State-of-the-art technologies	Page 03
2.3	Literature survey	Page 03

Chapter 3 System Requirements and Analysis

3.1	Domain and business requirements	Page 04
3.2	Customer-oriented requirements	Page 05
3.3	System function requirements	Page 06
3.4	System behavior requirements	Page 08
3.5	System performance and non-functional requirements	Page 09
3.6	System context and interface requirements	Page 11
3.7	Technology and resource requirements	Page 11

Chapter 4 System Design

4.1.	System architecture design	Page 12
4.2.	System data and database design (<i>for software project only</i>)	Page 13
4.3.	System interface and connectivity design	Page 13
4.4.	System user interface design (<i>for software project only</i>)	Page 13
4.5.	System component API and logic design (<i>for software project only</i>)	Page 19
4.6	System design problems, solutions, and patterns	Page 19

Chapter 5 System Implementation

5.1.	System implementation summary	Page 20
5.2.	System implementation issues and resolutions	Page 20
5.3.	Used technologies and tools	Page 21

Chapter 6 Development

6.1	Communication	Page 22
6.2	Timeline	Page 22
6.3	Team Contributions	Page 23

Chapter 7 System Testing and Experiment

7.1	Testing and experiment scope	Page 23
7.2	Testing and experiment approaches	Page 24
7.3	Testing report	Page 25

Chapter 8 Conclusion and Future Work

8.1	Project summary	Page 27
8.2	Future work	Page 28

References

[1]	NewsAPI	Page 29
[2]	Firebase	Page 29
[3]	React	Page 29
[4]	Lucidchart	Page 29

List of Figures

Figure 1. Activity diagram illustrates the process summary of the login system	Page 04
Figure 2. Use Case diagram illustrates the expected actions of site visitors seeking news	Page 04
Figure 3. User login process illustrated using State Machine Diagram	Page 09
Figure 4. Admin login process illustrated using State Machine Diagram	Page 10
Figure 5. System architecture demonstrated as an MVC (model-view-controller) architecture.	Page 12
Figure 6. System database functionality in UML notation	Page 13
Figure 7. The website UI when viewed as a guest	Page 13
Figure 8. The website UI when viewed as a user	Page 13
Figure 9. The website UI when viewed as an administrator	Page 13
Figure 10: The website UI of the image carousel on the home page.	Page 14
Figure 11: The website UI of Williamston news posts and public announcements on the home page.	Page 14
Figure 12. The website UI when visiting Michigan News page	Page 15
Figure 13. The website UI when visiting National News page	Page 15
Figure 14. The website UI for the archive page	Page 16
Figure 15. The website UI for the personal feed page	Page 16
Figure 16. The website UI for the edit page in admin view	Page 17
Figure 17. The website UI for the creating post in admin view	Page 17
Figure 18. The website UI for the deleting post in admin view	Page 17
Figure 19. The website UI for editing announcements in admin view	Page 18
Figure 20. The website UI for searching a news article through search bar	Page 18
Figure 21. Diagram represents the software component API and its functionality	Page 19
Figure 22. Diagram illustrates the software development timeline.	Page 22

List of Tables

Table 1.	List of researched product profiles and what was found	Page 02
Table 2.	List of expected user groups and their related use cases	Page 05
Table 3.	Description of Functional Requirements for the software system	Page 06
Table 4.	Description of Non-functional Requirements for the software system	Page 08
Table 5.	Identify the context environments supporting the development, testing, and deployment of the project results along with their descriptions.	Page 11
Table 6.	Description of each test criteria for component level	Page 23
Table 7.	Description of each testing criteria for system level	Page 24
Table 8.	Description of testing and experimental approaches	Page 24
Table 9.	Summary of software system testing report on Functional Requirements	Page 25
Table 10.	Summary of software system testing report on Non-functional Requirements	Page 26

Chapter 1. Introduction

1.1. Project goals and objectives

The project is to improve one aspect of the Williamston city website. The chosen aspect to improve on the website is its news offerings. There are specific requirements for the improvements, such as how the chosen focus on improving news posting requires features such as auto-archiving posts and an RSS feed. This project still allows for subjective interpretation of how requested features/functionalities will be implemented. A custom-designed user interface for displaying the news will be a large emphasis for this project, as it will drastically show the improvement from the previous website.

1.2. Problem and motivation

The Williamston city website is very outdated in both looks and functionality. The current website interface doesn't use the entire screen dimensions, and its main page that displays some city announcements remains cluttered. Residents must find it easy to read the text on the screen to stay up to date with the latest city news, so there should be an offering on the website that provides sections of organized and readable text in the UI.

1.3. Project application and impact

Williamston residents will be able to stay up to date with city events and public announcements along with the added functionality of attaining outsourced news from outlets directly from the website. By serving as a centralized hub for a diverse set of city, state, and even national news, the dissemination of crucial information is made convenient for Williamston residents. On top of this, an implementation of a RSS Feed empowers residents to stay connected with topics that matter to them. As a natural benefit, this application promotes civic engagement by informing and engaging residents of relevant news. A sense of community is also promoted by bringing residents together with news they may relate to. In all, this common digital platform fosters a collective understanding of local news or issues which further improves local engagement.

1.4. Project results and expected deliverables

Results: A website for the city of Williamston has been developed that has enhanced news options to fit the needs of their residents. New features include the auto archiving of Williamston-related news on the website homepage set for one month after the post dates. There are new dedicated pages for both Michigan-related and National-related news that source from an API to get the information displayed. An RSS Feed was an important feature that curates news based on preferential categories (sports, science, politics) based on the choices of logged in users.

Report: Written Project Report

Prototype: Functional website with newly implemented interface and expanded news features

Code: React.js code for designing the website with UI libraries and API data

Database: Firebase Realtime Database

Other: Firebase Authentication

1.5. Market research

Table 1: List of researched product profiles and what was found.

Researched	Findings
Similar Products	San Francisco City Website
Common Website Tools	React.js, HTML, CSS, Javascript, Firebase, Python, Flask, SQL

1.6. Project report Structure

1. Chapter 1 Introduction
2. Chapter 2 Project Background and Related Work
3. System Requirements and Analysis
4. System Design
5. System Implementation
6. Development
7. System Testing and Experiment
8. Conclusion and Future Work

Chapter 2 Background and Related Work

2.1. Background and used technologies

Many new feature implementations were specified in the Report for Proposal (RFP), and the requested changes to the News Posting features were chosen as the focus for enhancing the website. This is in conjunction with the request for an RSS Feed feature implementation for news on the website.

2.2. State-of-the-art

There are other city websites that also have pages for displaying local city news, with the similarities to this project being that the posts come from city administrators. The San Francisco City website has a small news section consisting of small headlines cards on its homepage. There is a button for seeing more news that leads to a section on the website that has all city news dating back to at least 6 months prior.

2.3. Literature survey

San Francisco City Website. (August 14, 2023). In sf.gov. Retrieved Sep 8, 2023, from <https://sf.gov/>

SF News Section. (August 14, 2023). In sf.gov. Retrieved Sep 8, 2023, from <https://sf.gov/news/all>

Chapter 3 System Requirements and Analysis

3.1. Domain and business Requirements

Activity (Process Summary) Diagram

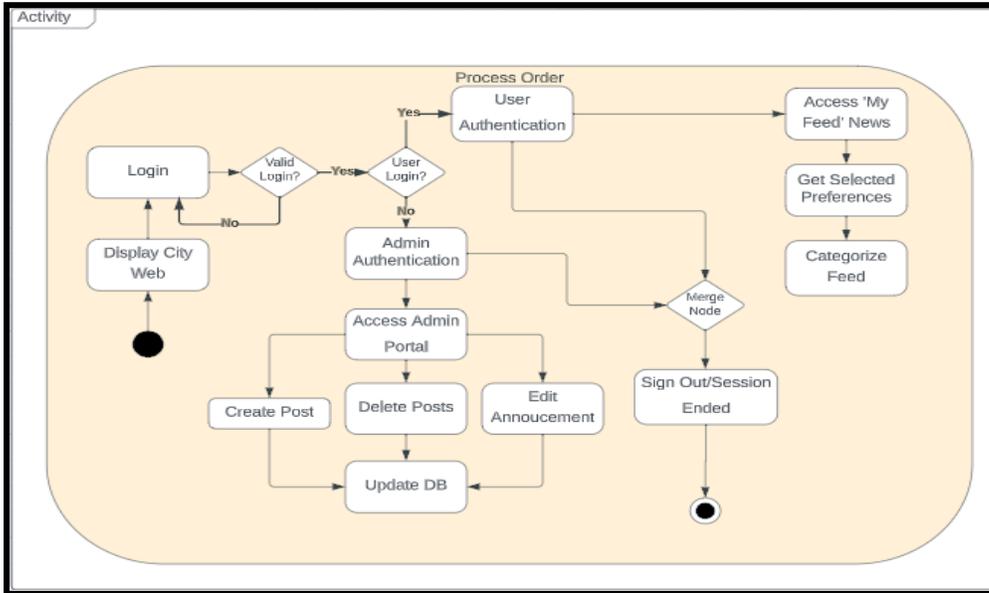


Figure 1: Activity diagram illustrates the process summary of the login system.

Visitor Use Case Diagram

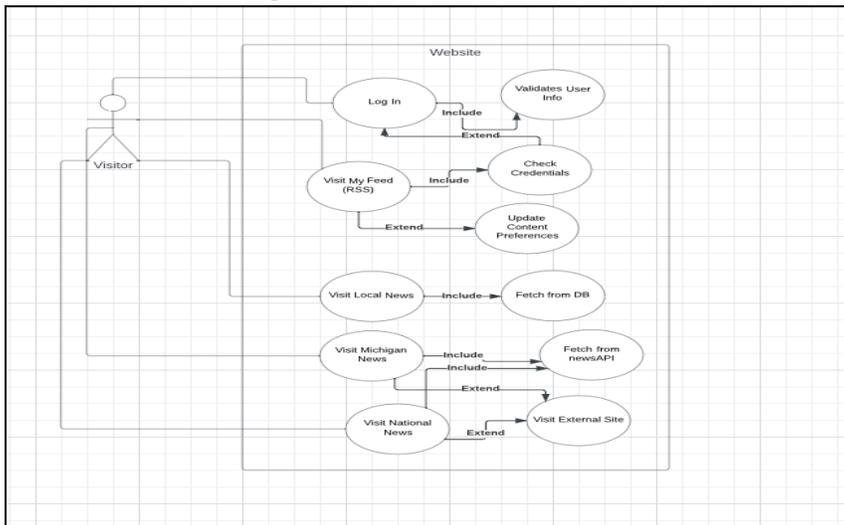


Figure 2: Use Case diagram illustrates the expected actions of site visitors seeking news

3.2. Customer-oriented requirements

Table 2: List of expected user groups and their related use cases

Users	Description of Use Case
Williamston City Residents	<ul style="list-style-type: none">- Main user group for revamped news portion of the website- They can stay informed about at least the city-specific news (public events and city advisories) on the homepage of the website for the news that heavily pertains to them- They can also access more general news about the state of Michigan and the entire US from outside news outlets- The news content can further be curated with the implementation of a custom RSS Feed that requires user login
City Administrators	<ul style="list-style-type: none">- Credentialed (with a log in) to edit the contents of the city news (Home) page of the website- Admin portal
Non-Resident Visitors	<ul style="list-style-type: none">- Website home page will also act as an introduction to prospective visitors- At the very top of the home page, right above city announcements, they can see a self-controlled slideshow of city monument and historic neighborhoods photographs

3.3. System function requirements

Table 3: Description of Functional Requirements for the software system.

Functional Requirement	Description
Williamston News page	<ul style="list-style-type: none"> - Writings of Williamston city news in announcements or posts about public events or city advisories can be seen on the main home page of the website. - Upon the home page loading, content is fetched from a database and rendered into web UI components
Auto Archive Posts	<ul style="list-style-type: none"> - City news posting shall only remain on the website for a short period of time (one month) so as to ensure that what is being displayed is still relevant. - That means that each individual post should be auto-archived by the system once it has been up for as long as the specified time period.
Broader News pages	<ul style="list-style-type: none"> - Additional news pages for outsourced state and national news take from the content fetched from newsAPI, and the content for those pages are also rendered on every page load. - Built-in auto-archiving as more recent news from NewsAPI will render first
User Login Verification	<ul style="list-style-type: none"> - Check that username and password entered match the records for a user in the database - Upon logging in, load user preferences for website news content on MyFeed Page
Administrative Credentials Verification	<ul style="list-style-type: none"> - Verify that login is the administrative username and password - Authenticated credentials through login will give access to hidden pages of the website to create posts and edit the announcement on the home page

Create/Delete Posts	<ul style="list-style-type: none"> - One page for writing posts to be shown prominently on the home page - Create Post page will have input fields for Article Title, Date, Author, and Text - Another page for deleting posts that will show snippets of the current posts visible on the website - A delete button next to each post snippet on the Delete Post page will remove the post from the home page once clicked upon
Announcements	<ul style="list-style-type: none"> - A page in the admin portal dedicated to editing the announcement section of the website - Text fields for date and text of announcement - The announcement section of the website, shown in a gray box to the right of city news posts, should show the most recently submitted announcement
Create User Account	<ul style="list-style-type: none"> - New users can sign up for an account on the website through a link from the regular login page - Check that password is at least 6 characters long
Maintain User Account	<ul style="list-style-type: none"> - Save selected categories from the checked boxes in the 'My Feed' page (unselected by default) - Every visit to the 'My Feed' page should have the same selected categories from before and those categories of news articles shown below the respective headers

3.4. System performance and non-functional requirements

Table 4: *Description of Non-functional Requirements for the software system.*

Non-Functional Requirement	Description
Fetch Reliability	In order for readers to see the news on the city news page specifically, the text and images must be successfully fetched from the database where the unarchived posts have been stored.
Content Correctness	The content written by city administrators that are to be displayed on the Williamston city news page must be stored in the database exactly as it was written on the administrator's end so that the intended text and images are being displayed to site visitors.
Website Performance	Contents of news pertaining to state and national news from NewsAPI should load within the time frame of 2 seconds so as to not have a noticeable delay. Content of city news should load within the similar 2 second time frame.
Content Security	There shall be no unauthorized access to the editing of the contents of the local news section, which is only meant to be edited by city administrators through an admin portal from secure login.
Website Security	Authentication of inputs from users to the website should prevent malicious attack through text fields that would allow access to website data

3.5. System behavior requirements

User Login State Machine Diagram

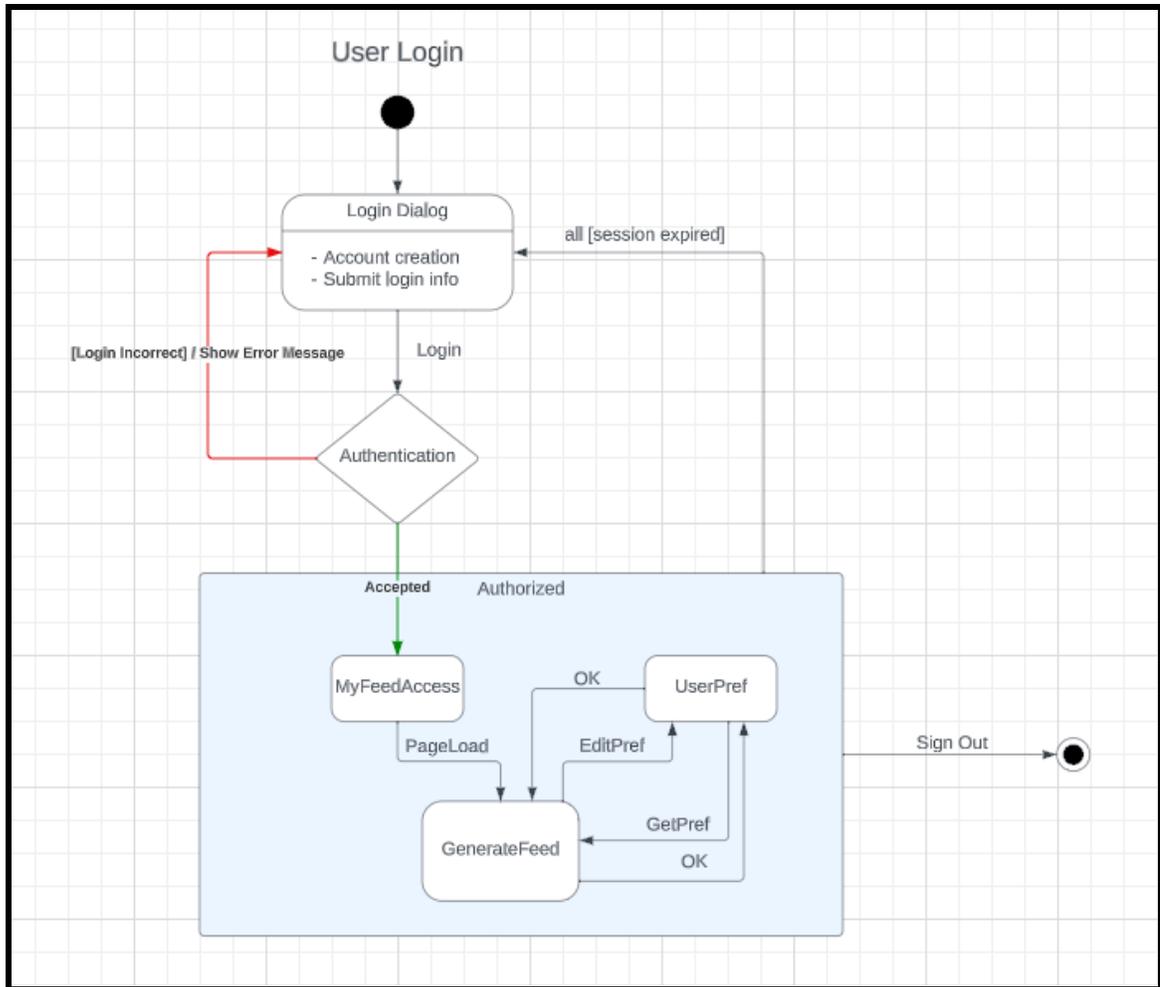


Figure 3: User login process illustrated using State Machine Diagram

Admin Login State Machine Diagram

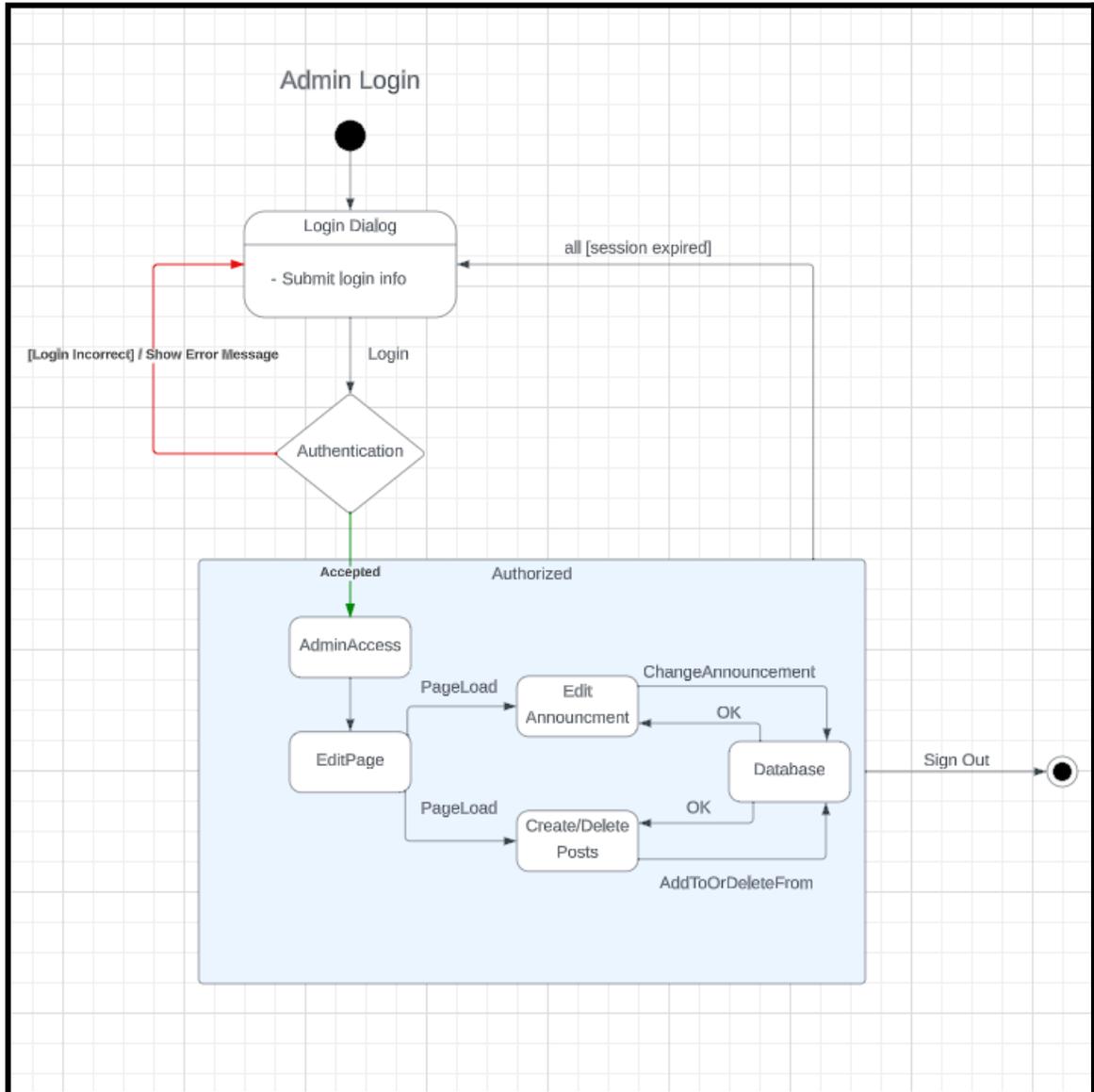


Figure 4: Admin login process illustrated using State Machine Diagram.

3.6. Context and interface requirements

Table 5: *Identify the context environments supporting the development, testing, and deployment of the project results along with their descriptions.*

Context	Description
User Interface Design	<ul style="list-style-type: none">- Top navigation bar on all pages for easy navigation to different pages of the website- Prominent text fields inputting information to the website with intuitive button placement- Text throughout the website should be easily readable in aspects of text color and size
Development	<ul style="list-style-type: none">- Aim build a functional website with the required and planned features- Utilize mainstream tools for development, such as with React and Firebase
Testing/Deployment	<ul style="list-style-type: none">- Confirm that the developed features work to expectation, accomplished by simulating usage of the website and checking that the different outcomes meet the specified functional and non-functional requirements- Check that the website runs/loads successfully on another computer, with previous tested features continuing to work to expectation

3.7. Technology and resource requirements

1. React.js (Java Library)
2. Material UI (React Component Library)
3. Firebase Realtime Database (Database)
4. Firebase Authentication (User Authentication service)
5. NewsAPI (Source of Outsourced News)
6. RSS Feed
7. Git/GitHub (Version Control System)
8. Vercel (Hosting)

Chapter 4 System Design

4.1. System architecture design

The system architecture for the final implementation is an MVC (model-view-controller) architecture which works well for designing a web-based application.

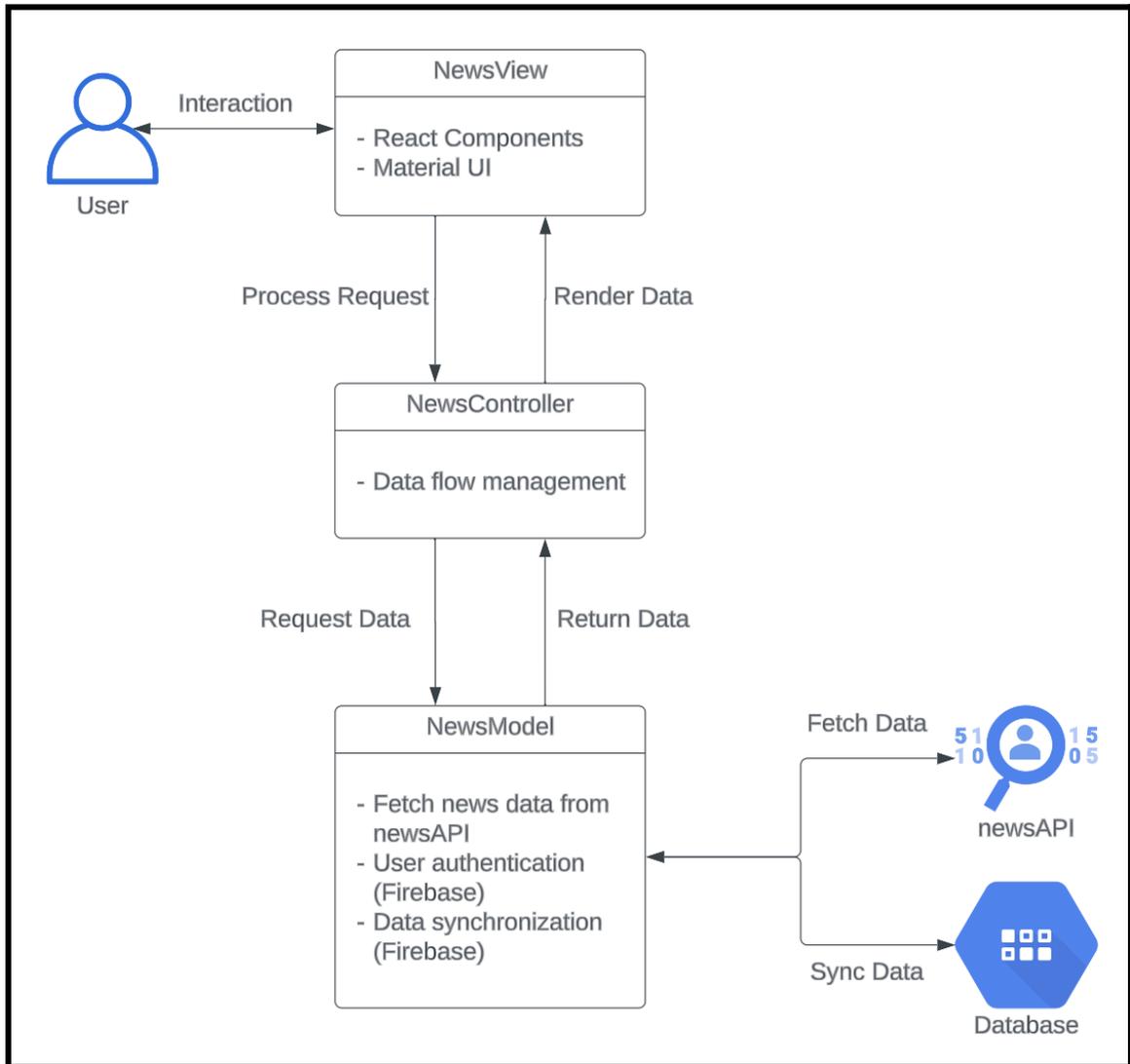


Figure 5: System architecture demonstrated as an MVC (model-view-controller) architecture.

4.2 System data and database design

Database Functionality



Figure 6: *System database functionality in UML notation.*

4.3 System interface and connectivity design

External Interfaces: Communication externally with NewsAPI to fetch news data from their publicly available API

Connectivity Protocol: Utilizing HTTPS, hyperlink transfer protocol with additional encryption and security for network communication

4.4 System user interface design (for software project only)

Guest UI



Figure 7: *The website UI when viewed as a guest.*

User UI



Figure 8: *The website UI when viewed as a user.*

Admin UI

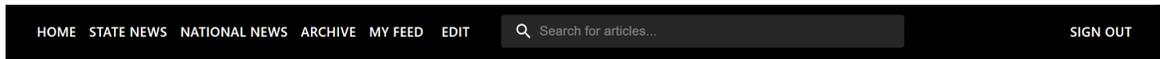


Figure 9: *The website UI when viewed as an administrator.*

Home Page

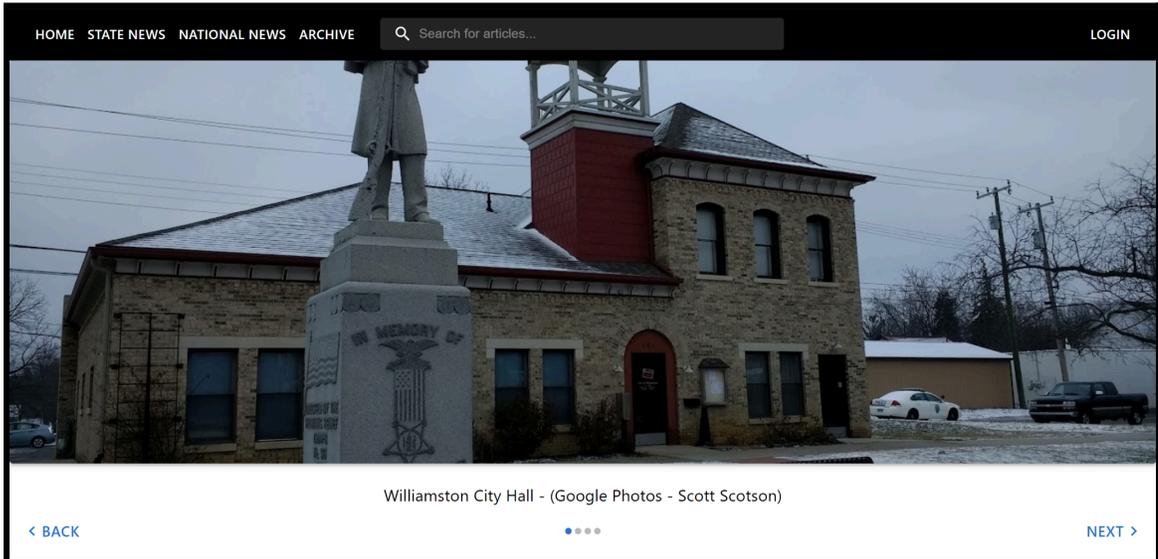


Figure 10: The website UI of the image carousel on the home page.

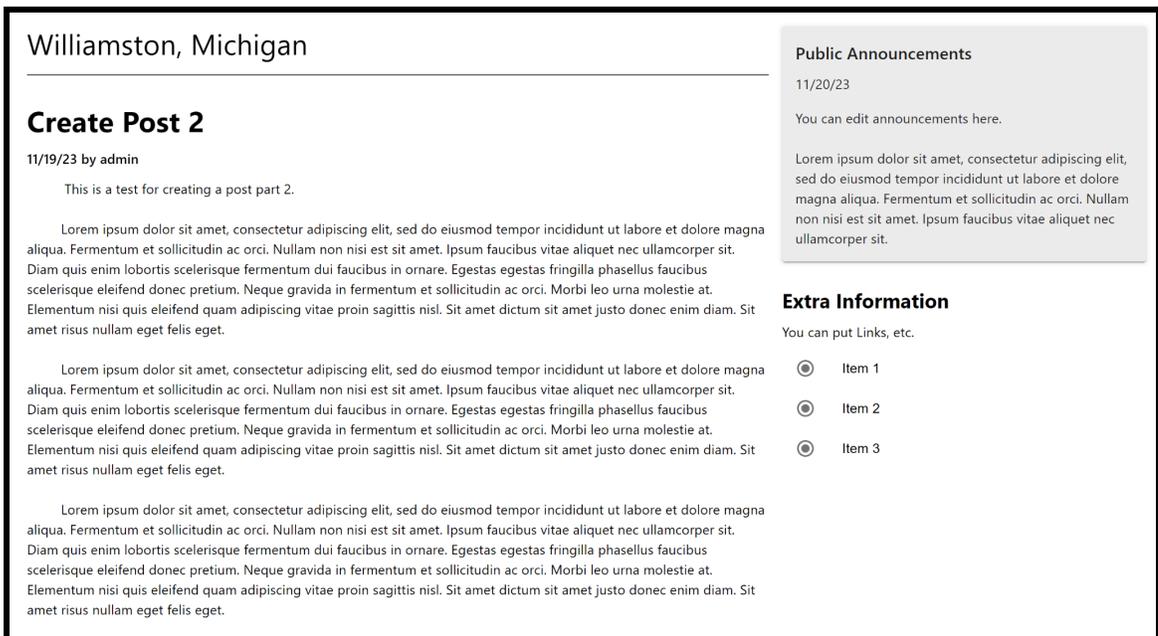


Figure 11: The website UI of Williamston news posts and public announcements on home page.

Michigan News

HOME STATE NEWS NATIONAL NEWS ARCHIVE Search for articles... LOGIN

In Michigan, where every vote counts, Arab Americans are turning away from Biden

The community turned out overwhelmingly for Joe Biden, helping him win the swing state. But many now feel betrayed by the president's response to the Israel-Hamas war and it could cost him in 2024.

Jeongyoon Han, NPR

READ MORE ...

Top Michigan Headlines

In Michigan, #RestoreRoe abortion rights movement hits its limit in the legislature

Michigan Radio, NPR

Last year, Michigan voters put the right to abortion in the state constitution. This year, the state legislature kept a 24-hour waiting period and said Medicaid can't pay for the procedure.

How chefs feed 100,000 Michigan fans at America's biggest football stadium

Abby Narishkin, Corey Isbell, Dorian Barranco, Business Insider

The University of Michigan's Big House is the biggest football stadium in the United States. In secret kitchens and tunnels, chefs make food for fans.

People in Missouri, Ohio, Michigan and other Midwestern states will have the highest heating bills this winter — here's why

Eliza Relman, Business Insider

Springfield, Missouri and Cleveland, Ohio will be impacted most by high heating bills, followed by cities across Missouri, Kansas, and Michigan.

Figure 12: The website UI when visiting Michigan News page.

US News

HOME STATE NEWS NATIONAL NEWS ARCHIVE MY FEED EDIT Search for articles... SIGN OUT

Israel-Hamas war: Live updates and latest news - The Associated Press

The Israeli military said the latest swap of hostages for Palestinian prisoners held by Israel is underway. Twelve Israeli hostages — nine women and a 17-year-old — as well as two foreigners were been handed over to the Red Cross in Gaza Tuesday evening. They...

READ MORE ...

Top US Headlines

Israel-Hamas war: Live updates and latest news - The Associated Press

The Associated Press, Associated Press

The Israeli military said the latest swap of hostages for Palestinian prisoners held by Israel is underway. Twelve Israeli hostages — nine women and a 17-year-old — as well as two foreigners were been handed over to the Red Cross in Gaza Tuesday evening. They...

Sean "Diddy" Combs Steps Down as Chairman of Revolt Amid Multiple Sexual Assault Lawsuits - Hollywood Reporter

Winston Cho, Hollywood Reporter

Two more women sued the music mogul for sexual assault last week after he settled a suit brought by R&B singer Cassie. He's denied the accusations.

"Geological hazards" lurking below Yellowstone National Park, data show - CBS News

Emily Mae Czachor, CBS News

New imaging technology has allowed geologists to see the bare earth, and its potential red flags, around Yellowstone like never before.

Figure 13: The website UI when visiting National News page.

Archives

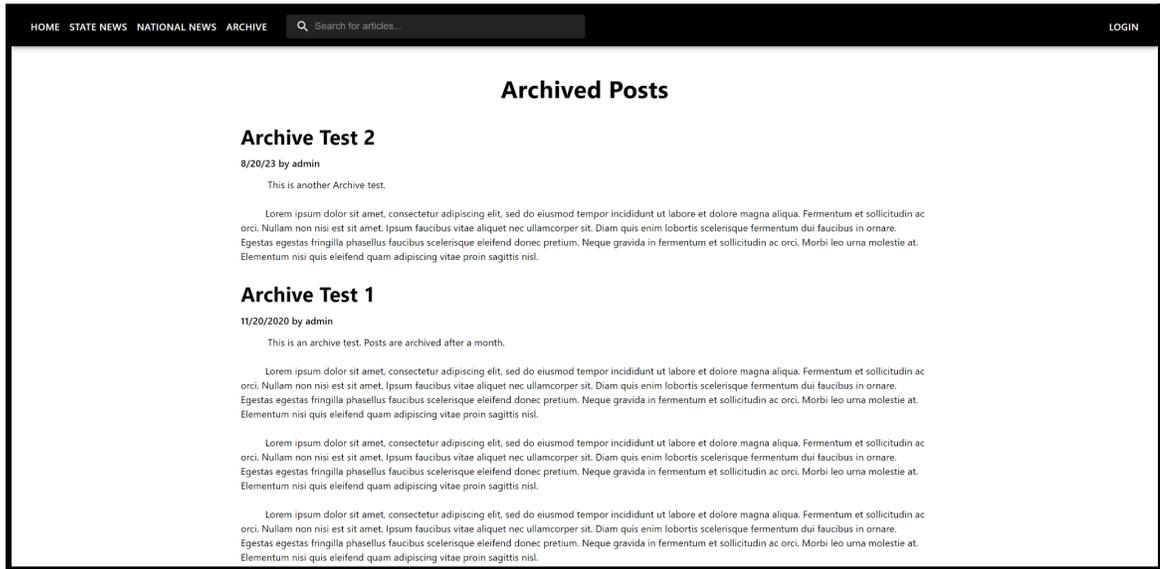


Figure 14: The website UI for the archive page.

My Feed

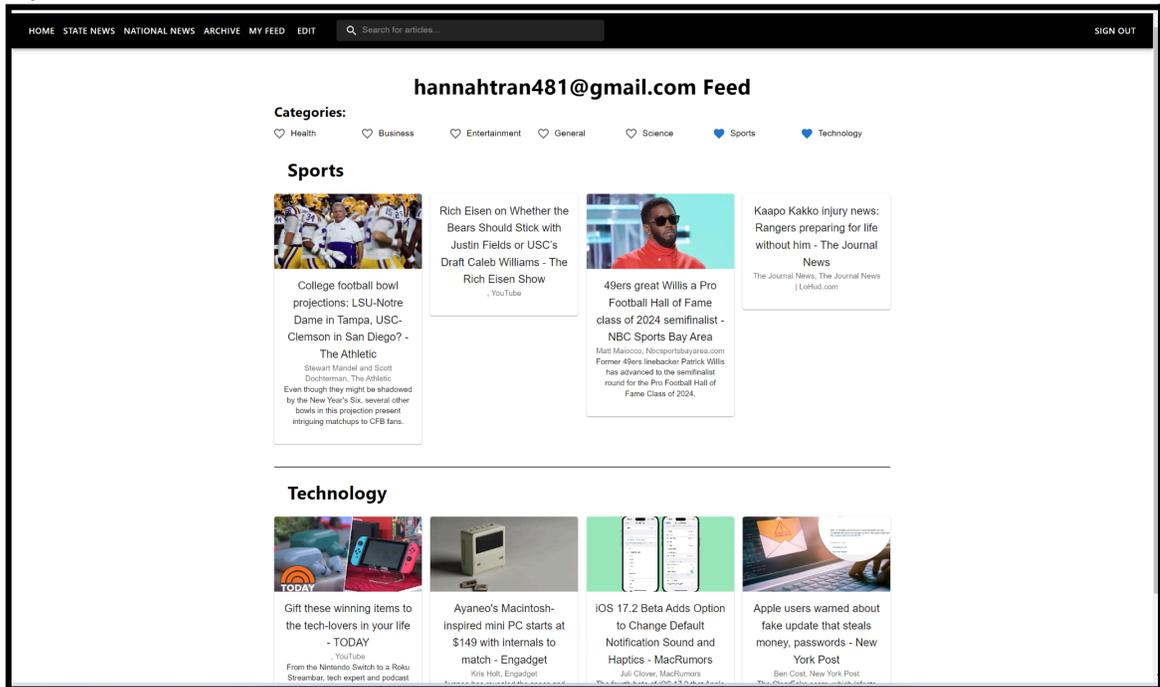


Figure 15: The website UI for the personal feed page.

Admin UI

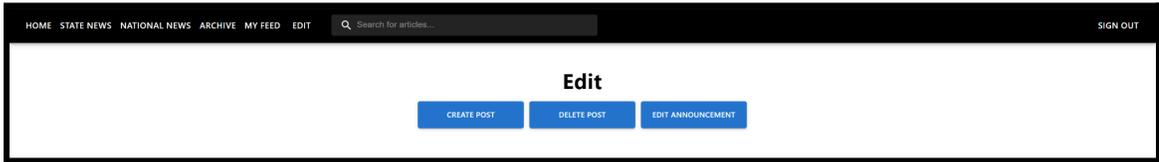


Figure 16: The website UI for the edit page in admin view.

Create Post UI

The image shows a form titled "Create Post" with a white background and a black border. The form contains four input fields, each with a label above it: "Article Title", "Date Posted", "Author", and "Input Body". Below the "Input Body" field, there is a blue button labeled "CREATE POST".

Figure 17: The website UI for creating a post in admin view.

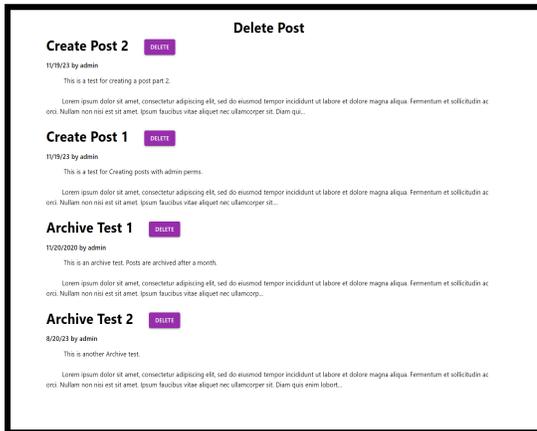


Figure 18: The website UI for deleting a post in admin view.

Announcement UI

Edit Announcement

Date

Announcement Info

[EDIT ANNOUNCEMENT](#)

Figure 19: The website UI for editing announcements in admin view.

Search

HOME STATE NEWS NATIONAL NEWS ARCHIVE

[LOGIN](#)

Mikel Arteta: Arsenal manager charged by FA for comments after Newcastle defeat

The Football Association charges Arsenal manager Mikel Arteta with misconduct for comments he made after his side's 1-0 Premier League defeat by Newcastle.

[READ MORE . . .](#)

Displaying articles for: football

Mikel Arteta: Arsenal manager charged by FA for comments after Newcastle defeat

, BBC News

The Football Association charges Arsenal manager Mikel Arteta with misconduct for comments he made after his side's 1-0 Premier League defeat by Newcastle.



Palestinian football's 'star defender' among players stuck in Gaza

<https://www.facebook.com/bbcnews>, BBC News

Ibrahim Abuimeir and two others won't be able to play in Tuesday's World Cup qualifier against Australia.



Figure 20: The website UI for searching a news article through search bar.

4.5. System component API and logic design

API Functionality

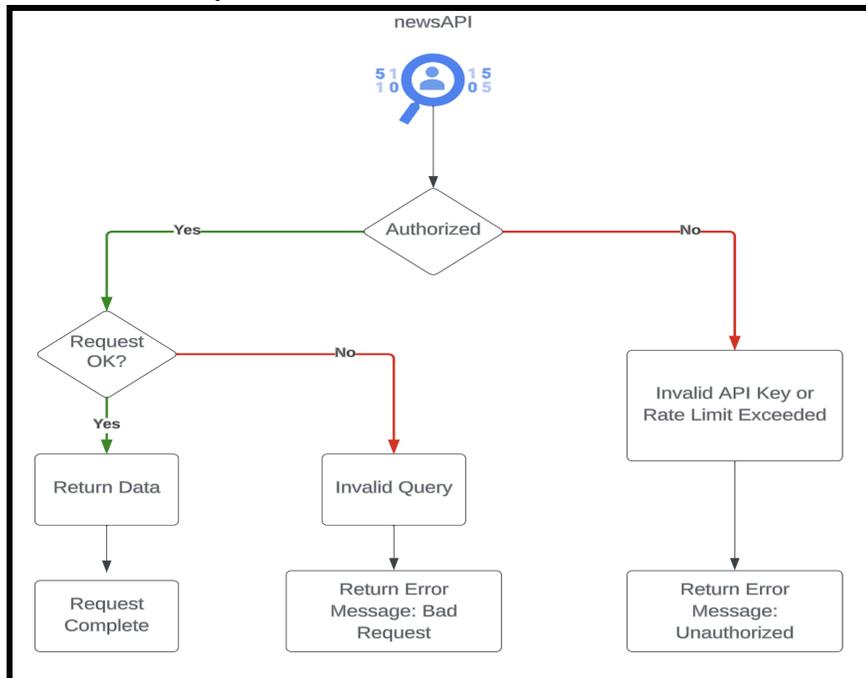


Figure 21: Diagram represents the software component API and its functionality

4.6. Design problems, solutions, and patterns

Initially, a component based architecture was used as it was React.js's most popular architecture usage. This presented some issues during further development because errors were hard to spot and debug within each component. Working with a component based architecture meant that each feature of the website was coded all together in separate components. This meant that data fetching, data rendering, and etc were all coupled together. With this in mind, the decision to switch to Model View Controller (MVC) was made to make the development process easier and more understandable for future code editors. MVC allows the website to be broken down into different areas to easier diagnose issues. If issues were confronted, it would be much easier to pinpoint if it is an issue with the data models, controllers, or rendering. Additionally, the MVC architecture facilitates scalability and adaptability through the easy integration of new features due to each independent component. Due to the time constraints, a design trade-off that was made was to simplify each system requirement to its basic usages. Complex enhancements were initially brainstormed before being realistically simplified. An instance of this in action is simplifying the admin UI to only three main features: edit posts, delete posts, and edit announcements. In the end, each requirement was met and functioned appropriately.

Chapter 5 System Implementation

5.1. System implementation summary

There is currently a working website that has the envisioned user interface. The main page of the website showcases local news postings. The admin portal that city administrators will use for writing those posts has been tested to ensure that new content successfully posts. Also, the state and national news pages are successfully utilizing content fetched from newAPI to populate text and images on their respective pages. Large clickable news cards have been implemented for a proper aesthetic to the website while adding extra functionality in being able to link to the outside news article.

5.2. System implementation issues and resolutions

During the development of this website, a variety of issues posed some challenges that required drastic changes. For instance, a big shift in implementation was made while converting the system architecture to fit the MVC mold. React.js typically utilizes component based architecture so altering the old initial implementation posed some challenges along the way. By employing MVC architecture, each component (model, view, controller) needed to cohesively work together to create the envisioned website. . Most of the errors stemmed from issues between connecting these components together so the resolution to this challenge was to decompose each feature in terms of the architecture needed to develop a deep understanding of the logic of each feature. During this process, architecture diagrams of this website were especially useful and a great solution. Another issue that was encountered was finding a hosting service for this website. Github Pages was initially used to host this website but it was later found that Github Pages did not support page routing for single page apps. This caused the website to produce 404 errors while trying to switch pages. To solve this issue, Vercel was used as another hosting alternative which worked smoothly. With this, the website shifted from Github Pages to Vercel. It is also important to note that the free version of NewsAPI only is able to provide its news fetching service under local hosting. Payment would be required for NewsAPI to display news on all domains so the website hosted on Vercel is unable to display any news from NewsAPI. As a group, the decision was made to hold off on any payments as this project is limited to the scope of the design only and the functionality of the full website could easily be shown through a local hosting.

5.3. Used technologies and tools

The user interface was created with the React.js library. Material UI, a react component library, was utilized to create the look and enhanced functionalities of the UI components. React is known to be a popular JavaScript library in the market so an interest to gain some experience in using this was considered. The coding was all done with the Visual Studio Code (VSCode) IDE as it is a code editor that supports several different languages. State and national news pages get their news from an API called NewsAPI, where the headline, summary text, and article image are displayed on the city website. NewsAPI was chosen as it had an easy implementation process and was free of charge in testing stages. Firebase services such as Firebase's realtime Database and Firebase Authentication were used as it simplified the logic behind handling authentication and provided an accessible database. Firebase Authentication was used to handle login/account creation while Firebase Database was used to store user preferences and admin privileges.

Chapter 6 Development

6.1. Communication

Communication was done mainly through a discord chat with the 4 team members. The free features of Discord, there were mainly voice calls held with the option to share/broadcast the current computer screen from one device in order to better complete the task at hand collaboratively. Meetings were regularly scheduled at least once every two weeks to ensure that the project was completed in a timely manner.

6.2. Timeline

Software Development Timeline

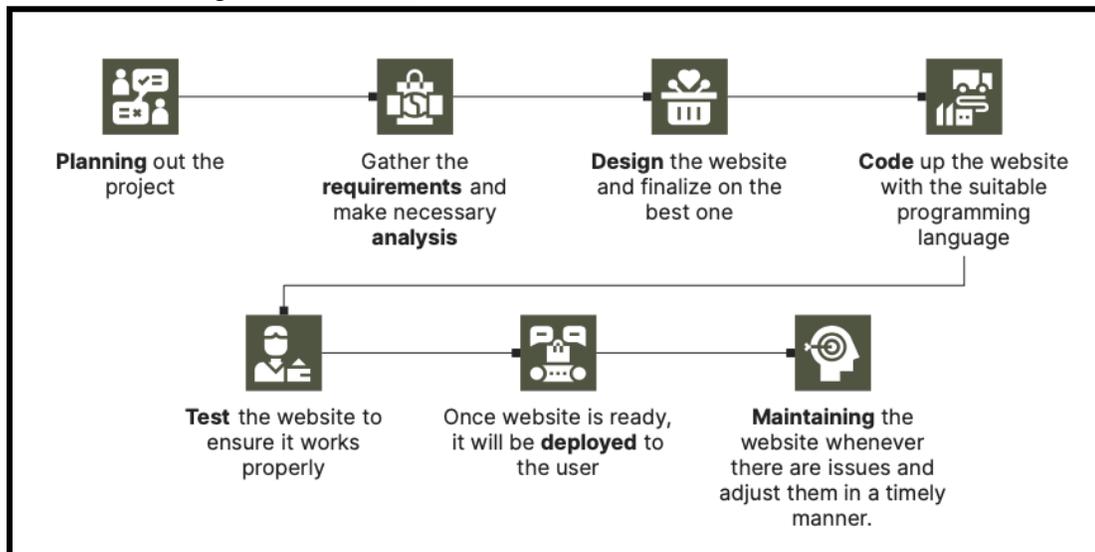


Figure 22: Diagram illustrates the software development timeline.

Starting Dates

Planning: 8/23/23

Requirements and Analysis: 9/22/23

Design: 9/23/23

Code: 10/01/23

Test: 11/21/23

Deployment: 11/25/23

Maintenance: 11/27/23

6.3. Contributions

Hannah Tran: Lead programmer for React, NewsAPI, and Firebase

Erickson Ng: API and React.js research, Project Report manager

Jason Huynh: Project Report contributions, Documentation

Viet Nguyen: Project Report contributions, Documentation

Chapter 7 System Testing and Experiment

7.1 Testing and experiment scope

1. Unit Testing:
 - a. Objective: Testing of individual components in isolation
 - b. Focus: Testing functionality of methods and objects based on project specifications
2. Integration Testing
 - a. Objective: Ensure that different software components are able to work together to obtain desired functionality and performance
 - b. Focus: Interaction between React components, Material UI Components, and Firebase services
3. System Testing:
 - a. Objective: Ensure the entire program fulfills functional and non-functional requirements
 - b. Focus: Verify that components are compatible, interact correctly and transfer the right data at the right time across their interfaces.

Component Level

Table 6: *Description of each test criteria for component level.*

Test Criteria	Description
Functionality	Each unit/component should fulfill its intended function without issues
Reliability	Components are able to stably and dependably operate under various stress conditions.
Error Handling	Components should be able to handle unexpected or error inputs
Code Coverage	Determine which components have not been tested

System Level

Table 7: *Description of each testing criteria for system level.*

Test Criteria		Description
Functional Requirements		Verify that all of the system's functional criteria are met.
Non-Functional Requirements	Performance	Evaluate system's response time and load handling under various conditions
	Security	Identify and resolve possible security vulnerabilities
	Maintainability	Evaluate the difficulty of performing bug fixes and system updates
	Availability	Ensure minimal downtime and disruptions to system performance

7.2 Testing and experiment approaches

Table 8: *Description of testing and experimental approaches.*

Functional Requirements		Items are automatically archived by the system based on predefined criteria. RSS feed generates and updates in real-time with most recent news items.
Non-Functional Requirements	Performance	Pages load in a timely manner within a second or two, including the pages which fetch large batches of data from the external news API
	Security	User authentication and login data is enforced and encrypted
	Maintainability	Ease of bug fixes and system updates Identify bugs and rapid update Communication with customers for rapid changes
	Availability	Simulate a variety of downtime scenarios to assess the system's recovery mechanisms and how it handle

7.3 Testing and experiment

Table 9: *Summary of software system testing report on Functional Requirements.*

Functional Testing	Pass/No Pass	Description
Williamston News page	Passed	Website was able to write Williamston city news in announcements and posts about public events or city advisories can be seen on the main home page of the website. Content was fetched from a database and rendered into web UI components
Auto Archive Posts	Passed	City news posts that are more than one month old are now in the archive. This demonstrated the system auto archive on one month old posts which ensured the relevancy in posts' content.
Broader News pages	Passed	Website was able to display content from state and national news which it fetched from NewsAPI.
User Login Verification	Passed	It checked the username and password to see if it matched with the database. It asked the user to input the username and password again if input doesn't match the database.
Administrative Credentials Verification	Passed	The system verified the administrative username and password which showed a hidden page that allowed administrators to create and edit announcements in the home page.
Create/Delete Posts	Passed	The website demonstrated its functionality for creating and deleting posts. When creating a post, it showed the expected input fields for Article Title, Date, Author, and Text. For deleting a post, it successfully deleted a post when clicked on the delete post button next to the respective post.

Announcements	Passed	The website did show a page which is dedicated to editing the announcement section. Users were able to edit and publish the announcement which was displayed as expected.
Create User Account	Passed	Successfully signed up for an account on the website. In addition, the website checked for passwords that need to be at least 6 characters long as expected.
Maintain User Account	Passed	The website saved selected news categories into My Feed page. It displayed news from selected categories when opening the My Feed page.

Table 10: *Summary of software system testing report on Non-functional Requirements.*

Non-Functional Testing	Pass/No Pass	Description
Fetch Reliability	Passed	The system fetched text and images from the database which displayed the news content successfully.
Content Correctness	Passed	The content was shown correctly as compared to how it was written on the administration's end.
Website Performance	Passed	The website was able to load city, state and national news from the NewsAPI within a 2 seconds time frame.
Content Security	Passed	The website was able to avoid any unauthorized access to editing any contents from the local news section through login verification.
Website Security	Passed	With the implementation of authentication of inputs, the website prevented any malicious attack through text fields that can enable access to website data.

Chapter 8 Conclusion and Future Work

8.1 Project summary

Source Code: <https://github.com/Confusedin131/Webpage>

Hosted Website: <https://newsapp-red.vercel.app> (Please refer to Section 4.4 for visual aspects of the website that rely on NewsAPI, such as Michigan News, National News, and the search news functionality, as usage of the API on non-testing domains including the hosted website above would require payment)

Status: A functioning website has been successfully developed. All the requested features chosen from the RFP (request for proposal) have been implemented and tested to meet the requirements of the project.

Results: City administrators now have a credentialed admin portal where they can edit the news postings on the homepage of the new Williamston city website. Those news postings can be seen by visitors on the homepage subsequently after the updates are sent to the Firebase server where the contents of each post are stored. Those posts directly from the city will now auto-archive exactly one month after the original post date. There are also dedicated pages for Michigan news along with one for National news that present news in organized cards with text and photo data taken from an outsourced news API. Additionally, while not one of the required features for the website, a search feature was implemented that queries the external API for news based on user input into a persistent search bar. Finally, the RSS Feed feature on the website now allows for curated content for login users, where users who have created an account and are logged in have access to a ‘My Feed’ page that has options to choose which categories of news they would like to see on that page.

Experience and Lessons: Design wise, this project helped us understand the application of design principles and its processes in creating a scalable website. Lots of planning and considerations of design patterns were needed in shaping the blueprints for implementation which later provided us with benefits. With a solid understanding of the underlying architecture for the website, developing the website was a lot easier which exemplifies the importance of such processes. Coding wise, the implementation of this news website for Williamston helped us gain more experience and skills in React.js as most of us are unfamiliar with it. It was a new experience working with a diverse set of tools such as APIs, databases, authentication services, and libraries. While implementing the required features from the design, valuable insights of breaking down each feature into its core components to easily pinpoint the errors were found. The importance of understanding each component’s role in a broader sense of the system alongside the comprehension of each component as a single module was also discovered as well.

8.2 Future work

There is always room for improvement with a website, especially with how many features were available to be chosen for improving the Williamston website on the RFP. Future works may include adding sections of the website that give more general information about the city Williamston such as an about us and/or contact us page. Another area of interest could be to improve the security behind the webpage.

References

Documentation. News API. (n.d.). <https://newsapi.org/docs>

Google. (n.d.). *Firebase documentation.* Google. <https://firebase.google.com/docs>

React Documentation. React. (n.d.). <https://react.dev/learn>

UML Diagram Software. Lucidchart. (n.d.).

<https://www.lucidchart.com/pages/landing/uml-diagram-software>