

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Weather App

Team Members: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Product Owner(s): Nelson De Jesus Saroza Camacho

Mentor(s):

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document provides a comprehensive overview of the development process for a weather application, designed to deliver users accurate and ad-free weather information. The project was motivated by the need to offer reliable weather data without the interruptions or costs associated with many existing weather services. The development process entailed implementing user scenarios utilizing React for the frontend interface and integrating a dependable weather API for accurate data retrieval. Following methodologies, the project included thorough sprint plans and review sessions to ensure continuous enhancement and alignment with user requirements. The systems architecture emphasized modularity, incorporating industry practices in UI/UX design and data handling. This document also includes system validation processes and appendices with UML diagrams, UI designs, and sprint review reports, highlighting the iterative and collaborative nature of the project.

Introduction.....	5
Current System.....	5
Purpose of New System.....	5
User Stories.....	6
Implemented User Stories.....	6
Pending User Stories.....	7
Project Plan.....	8
Hardware and Software Resources.....	8
Sprints Plan.....	9
Sprint 1: Project Setup and Initial Research.....	9
Sprint 2: UI Design and Basic Component Implementation.....	9
Sprint 3: API Integration and Data Handling.....	9
Sprint 4: Testing and Refinement.....	10
Sprint 5: Documentation and Finalization.....	10
System Design.....	10
Architectural Patterns.....	10
System and Subsystem Decomposition.....	11
Deployment Diagram.....	11
Design Patterns.....	11
System Validation.....	12
Glossary.....	13
Appendix.....	15
Appendix A - Sprint Review Reports.....	15

INTRODUCTION

Nature shapes our everyday lives. Staying informed about the weather is a very important concept. It allows us to become prepared to face the day. Not only that but weather knowledge can help us prevent loss of life especially in natural phenomena. Storm season has arrived in Florida and the eastern coast of the US. People should be informed of the weather at any moment's notice. Staying informed in these situations can be very crucial in getting through them. Staying informed about the weather is helpful in any part of the world, this app will tell the weather based on, whether it's a hurricane in the east coast, tornadoes in the middle of the country or heatwaves, knowledge of the weather is a vital part of daily lives.

Current System

While other weather apps do provide information, they often ask the user to pay extra or subscribe to keep using their services, sometimes even omitting the amount of data they display to the free users. Other websites may also overwhelm the user with annoying ads. Most of the time users will have to navigate between different pages to obtain the information they need. Some of those ads may even discourage users from interacting with the app or website altogether.

Purpose of New System

Our team believes that weather information should be readily available on demand. Compared to other apps and websites that require users to subscribe to view complete weather information, our app is different. Our weather app provides easy to understand weather information without having to watch any advertising or other distractions. Unlike other weather apps that ask the user to subscribe to unlock sometimes basic weather information. Our app will provide that information in a very simple to read package. It does not include any ads or subscription screens, and all the info will be available from the moment they select what location they need to see.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the "Weather App" project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

#001: As a team member, I will research and evaluate the usage of React for building web applications, including its features, advantages, and best practices.

#002: As a team member, I will research weather APIs and evaluate their suitability for our app's requirements, considering factors such as data accuracy, coverage, and cost.

#003: As a team member, I will collaborate with team members to set up the project environment using Create React App, including configuring dependencies, folder structure, and version control with Git.

#004: As a team member, I will explore and experiment with React basics, such as components, props, and state management, through tutorials and hands-on exercises.

#005: As a team member, I will participate in brainstorming sessions to sketch out the user interface design for the weather app, considering user experience and visual aesthetics.

#006: As a team member, I will contribute to the development of UI components based on the agreed-upon design, using React and CSS frameworks like Material-UI or Bootstrap.

#007: As a team member, I will collaborate with designers and front-end developers to ensure consistency in UI components and adherence to design mockups.

#008: As a team member, I will integrate user feedback and iterate on UI components as necessary to improve usability and accessibility.

#009: As a team member, I will research and select a weather API provider based on the findings from the previous sprint, considering factors like data accuracy, reliability, and ease of integration.

#010: As a team member, I will implement utility functions or modules to handle API requests and responses, ensuring proper error handling and data parsing.

#011: As a team member, I will integrate the chosen weather API into our app and set up endpoints for fetching weather data based on user input or location.

#012: As a team member, I will work on the backend to establish data models and schemas for storing and processing weather data efficiently.

#013: As a team member, I will develop and integrate features such as current weather display, search functionality, and weather forecasts into the app, ensuring they meet acceptance criteria.

#014: As a team member, I will write unit tests for critical app functionalities and conduct integration tests to ensure proper functionality and reliability.

#015: As a team member, I will perform user acceptance testing (UAT) and address any issues or bugs identified during testing.

#016: As a team member, I will gather feedback from users to understand their needs and improve the app iteratively.

#017: As a team member, I will create documentation that includes the app architecture, codebase, and APIs.

#018: As a team member, I will create a poster highlighting the work done in our project.

#019: As a team member, I will finalize my poster for our project based on feedback from the product owner.

Pending User Stories

None.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- OpenWeatherMap
- WhatsApp
- GitHub
- Python
- Replit

Sprints Plan

Sprint 1: Project Setup and Initial Research

- **Objective:** Establish the project environment and conduct initial research.
- **Tasks:**
 - Research React and weather APIs.
 - Set up project repository on GitHub.
 - Configure project environment using Create React App.
 - Install necessary dependencies and set up version control.
- **Deliverables:**
 - Project repository with initial setup.
 - Research documentation on React and weather APIs.

Sprint 2: UI Design and Basic Component Implementation

- **Objective:** Design the user interface and implement basic components.
- **Tasks:**
 - Sketch initial UI designs considering user experience.
 - Implement foundational UI components using React.
 - Apply CSS frameworks (e.g., Material-UI) for styling.
- **Deliverables:**
 - UI design mockups.
 - Basic React components for the weather app.

Sprint 3: API Integration and Data Handling

- **Objective:** Integrate the weather API and handle data fetching and display.
- **Tasks:**
 - Select and integrate a suitable weather API.
 - Implement utility functions for API requests and error handling.
 - Develop features for displaying current weather and forecasts.
- **Deliverables:**
 - Integrated weather API with functional data fetching.
 - Display of weather information in the app.

Sprint 4: Testing and Refinement

- **Objective:** Conduct testing and refine the application based on feedback.
- **Tasks:**
 - Write and execute unit and integration tests.
 - Perform user acceptance testing (UAT) and gather feedback.
 - Refine UI and functionalities based on test results and user feedback.
- **Deliverables:**
 - Test reports and bug fixes.
 - Refined and improved weather app.

Sprint 5: Documentation and Finalization

- **Objective:** Finalize the project and prepare documentation.
- **Tasks:**
 - Complete project documentation, including architecture, codebase, and API usage.
 - Create a project poster summarizing key features and outcomes.
 - Conduct final review and prepare for project presentation.
- **Deliverables:**
 - Comprehensive project documentation.
 - Project poster.
 - Ready-to-present weather app.

System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

The system employs a modular architecture to separate concerns and enhance maintainability. The architecture includes:

- **Client-Server Architecture:** The client-side is built with React, providing a responsive and interactive user interface. The server-side handles data requests and processing.
- **Modularity:** Each component, such as data fetching, UI rendering, and state management, is developed as an independent module.
- **RESTful API Integration:** The app interacts with the OpenWeather API using RESTful principles, ensuring scalable and stateless communication.

System and Subsystem Decomposition

The system can be decomposed into the following subsystems:

- **Data Collection:** Modules responsible for fetching weather data from the OpenWeather API.
- **Data Processing:** Functions for parsing and handling data, including error handling and data transformation.
- **UI Components:** React components for displaying weather information, user input forms, and navigation elements.
- **State Management:** Utilization of React hooks and context for managing the application state and data flow.

Deployment Diagram

The deployment involves a client-server model:

- **Client-Side:** Deployed on users' web browsers. The React application runs locally, interacting with the server via HTTP requests.
- **Server-Side:** Hosted on a cloud platform, responsible for handling API requests, processing data, and serving responses.

Design Patterns

The following design patterns are employed:

- **Model-View-Controller (MVC):** Separates data (Model), user interface (View), and control logic (Controller) to enhance modularity and scalability.
- **Observer Pattern:** Ensures real-time updates to UI components when data changes.
- **Factory Method:** Dynamically creates instances of weather components based on user interactions and API responses.

SYSTEM VALIDATION

System validation involves multiple stages to ensure functionality and reliability:

- **Unit Testing:** Each component and function is tested in isolation to verify correctness.
- **Integration Testing:** Tests the interaction between different modules, ensuring seamless data flow and communication.
- **User Acceptance Testing (UAT):** Engages real users to test the app in real-world scenarios, providing feedback on usability and performance.
- **Bug Tracking and Resolution:** Identifies and fixes issues based on test results and user feedback.

GLOSSARY

- **User Interface (UI):** The graphical interface through which users interact with a software application. In the context of this project, the UI allows users to visualize crime data, filter information, and access insights generated by the system.
- **Web-Based Application:** An application that is accessed and used through a web browser, typically hosted on a remote server and accessed over the internet. In the context of this project, the crime analysis system is implemented as a web-based application, allowing users to access it from any device with an internet connection.
- **Sprint:** In agile software development, a sprint is a time-boxed iteration during which a specific set of tasks or features is completed. Sprints typically last for 1-4 weeks and end with a review of the work completed during the sprint.
- **OpenWeather API:** An API provided by OpenWeatherMap that supplies weather data, including current conditions, forecasts, and historical data. It is used in this project to fetch weather information for different locations.
- **React:** A JavaScript library for building user interfaces, maintained by Facebook and a community of developers. React allows for the creation of reusable UI components and efficient management of application state.
- **Component:** A modular, reusable piece of UI in a React application. Components can be nested, managed, and handled independently, promoting maintainability and scalability in the application.
- **State Management:** The practice of managing the state of an application, ensuring that data flows seamlessly between components. In React, state management can be handled using React hooks or state management libraries like Redux.
- **API Integration:** The process of connecting and interacting with external APIs within an application. In this project, API integration involves fetching weather data from the OpenWeather API and displaying it in the app.
- **Unit Testing:** A software testing method where individual components or functions are tested in isolation to verify their correctness. Unit tests are essential for ensuring the reliability of each part of the application.
- **Integration Testing:** A testing phase where multiple components or systems are tested together to ensure they work correctly in combination. This type of testing verifies the interactions and data flow between different parts of the application.
- **User Acceptance Testing (UAT):** A testing process where real users evaluate the application to ensure it meets their needs and requirements. UAT provides valuable feedback on usability and functionality from the end-user perspective.

- **Modularity:** A design principle that divides a system into smaller, self-contained modules. Each module is responsible for a specific functionality, making the system easier to manage, maintain, and scale.
- **Client-Server Architecture:** A network architecture where the client (frontend) interacts with the server (backend) to request and retrieve data. In this project, the React frontend communicates with the server to fetch weather data from the OpenWeather API.
- **RESTful API:** An API that adheres to the principles of Representational State Transfer (REST). It uses standard HTTP methods and status codes, allowing for scalable and stateless communication between client and server.
- **Error Handling:** The process of anticipating, detecting, and resolving errors or exceptions in a software application. Proper error handling ensures that the application can gracefully recover from unexpected issues and provide meaningful feedback to the user.
- **Feedback Loop:** A process where user feedback is continuously gathered and used to make improvements to the application. This iterative process helps refine features and enhance user satisfaction.

APPENDIX

Appendix A - Sprint Review Reports

Sprint 1: Review Meeting Minutes

Attendees: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Start time: 5:30 pm

End time: 6:00 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- #001: As a team member, I will research and evaluate the usage of React for building web applications, including its features, advantages, and best practices.
- #002: As a team member, I will research weather APIs and evaluate their suitability for our app's requirements, considering factors such as data accuracy, coverage, and cost.
- #003: As a team member, I will collaborate with team members to set up the project environment using Create React App, including configuring dependencies, folder structure, and version control with Git.
- #004: As a team member, I will explore and experiment with React basics, such as components, props, and state management, through tutorials and hands-on exercises.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None

Sprint 2: Review Meeting Minutes

Attendees: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Start time: 5:30 pm

End time: 6:00 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- #005: As a team member, I will participate in brainstorming sessions to sketch out the user interface design for the weather app, considering user experience and visual aesthetics.
- #006: As a team member, I will contribute to the development of UI components based on the agreed-upon design, using React and CSS frameworks like Material-UI or Bootstrap.
- #007: As a team member, I will collaborate with designers and front-end developers to ensure consistency in UI components and adherence to design mockups.
- #008: As a team member, I will integrate user feedback and iterate on UI components as necessary to improve usability and accessibility.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None

Sprint 3: Review Meeting Minutes

Attendees: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Start time: 5:30 pm

End time: 6:00 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- #009: As a team member, I will research and select a weather API provider based on the findings from the previous sprint, considering factors like data accuracy, reliability, and ease of integration.
- #010: As a team member, I will implement utility functions or modules to handle API requests and responses, ensuring proper error handling and data parsing.
- #011: As a team member, I will integrate the chosen weather API into our app and set up endpoints for fetching weather data based on user input or location.
- #012: As a team member, I will work on the backend to establish data models and schemas for storing and processing weather data efficiently.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None

Sprint 4: Review Meeting Minutes

Attendees: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Start time: 2:15 pm

End time: 2:45 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- #013: As a team member, I will develop and integrate features such as current weather display, search functionality, and weather forecasts into the app, ensuring they meet acceptance criteria.
- #014: As a team member, I will write unit tests for critical app functionalities and conduct integration tests to ensure proper functionality and reliability.
- #015: As a team member, I will perform user acceptance testing (UAT) and address any issues or bugs identified during testing.
- #016: As a team member, I will gather feedback from users to understand their needs and improve the app iteratively.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None

Sprint 5: Review Meeting Minutes

Attendees: Victor Ills Fernandez, Emmanuel Mejia-Galvan, Alejandro Andres Pazos, Nelson De Jesus Saroza Camacho, and Gabriela Smit

Start time: 5:30 pm

End time: 6:00 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- #017: As a team member, I will create documentation that includes the app architecture, codebase, and APIs.
- #018: As a team member, I will create a poster highlighting the work done in our project.
- #019: As a team member, I will finalize my poster for our project based on feedback from the product owner.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None