Software Requirements Specification

for

Weather Web Application

Version 1.0 approved

Prepared by Pratibha Yadav

Galgotias College of Engineering & Technology

14 February 2025

Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this SRS document is to outline the requirements for the system. The document will describe how the software will display local weather data and analyze the weather forecast. This product will keep us informed about conditions including temperature, rain, wind etc.

1.2 Document Conventions

The product is named WebCast. It will be able to collect and display local weather data, analyse weather forecast and display a live stream of the local weather. The product will be able to predict the weather. The product will be accessible through a website where user will have the benefit of seeing what the weather looks at a particular location. The main goal is to have a website that will provide user information about the weather.
We can use the following: -

1. *Automated process e-alert/warning*
2. *Creation of infrastructure facilities*
3. *Service for Providing Information on Forecasted Weather & Agro-Met Advisory Services*
4. *Provide government services.*

1.3 Intended Audience and Reading Suggestions

The intended audience for this software requirement specification includes all individuals involved in the creation, development, and operation of the WebCast weather application. This encompasses stakeholders, software developers, project supervisors, and validation teams. The application is designed for a wide range of users, such as students, office workers, business owners, or anyone whose daily activities might be impacted by sudden changes in weather forecasts.

For readers seeking an overview of the product, the ***Introduction*** *section provides important insights. To gain a more in-depth understanding of the application's functionality and design,* ***Chapter 2*** *is the appropriate reference. Information regarding system-user interfaces and the software-hardware implementation can be found in the* ***External System Interface*** *section. Additionally, the* ***Non-Functional Requirements*** *section outlines the non-technical criteria for evaluating the system's performance, offering a broader perspective on its functions and operations. This document serves as a guide for those involved in building, managing, and using the WebCast app, ensuring clear communication and understanding across all parties.*

1.4 Project Scope

The WebCast weather application is designed to collect and analyse weather data, providing users with real-time updates and forecasts to assist in their daily activities. It offers details such as wind speed, humidity, temperature, and the overall weather conditions for the day. The system displays both hourly and daily weather changes, while also providing a weekly forecast to help users plan ahead. Additionally, WebCast streams a live map that updates with changing weather patterns. The app can automatically detect the user's location, but users also have the option to manually set or search for locations to get specific weather data. Users can receive live updates via notifications and customize the app by selecting their preferred weather units. The app also allows users to set alerts for severe weather events, providing them with timely warnings. WebCast helps users make informed decisions by offering detailed forecasts, ensuring they can adjust their plans based on expected weather conditions throughout the day.

1.5 References

1. *Peng, Dunlu & Cao, Lidong & Xu, Wenjie. (2011). Using JSON for Data Exchanging in Web Service Applications. 7.*
2. *Garegnani, G., Hencken, K., & Kassubek, F. (2024). A Physics-Inspired and Data-Driven Approach for Temperature-Based Condition Monitoring. PHM Society European Conference, 8(1), 10.*
3. *OpenWeatherMap API by https://openweathermap.org/api*
4. *Mozilla Developer Network. Fetch API. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API. 2023*
5. *Mozilla Developer Network. Async function. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function 2023*

2. Overall Description

2.1 Product Perspective

WebCast is a new, self-contained weather application developed using HTML, CSS, and JavaScript. It is designed to provide real-time weather data and forecasts to users, offering features such as location detection, notifications, and customizable settings. This application operates independently and does not rely on any existing systems or product families.

2.2 Product Features

The WebCast will gather and show the nearby climate information using API. The application will likewise have a connection to a climate site where it will show the present temperatures alongside a 5-day forecast.
WebCast offers several key features to enhance user experience:

1. ***Real-time Weather Updates**: Provides current weather data, including temperature, humidity, wind speed, and day type.*
2. ***Hourly and Daily Forecasts**: Displays weather changes over hours and days, helping users plan ahead.*
3. ***Weekly Forecast**: Offers a weekly weather outlook for advanced planning.*
4. ***Location Detection**: Automatically detects the user's location or allows manual entry of locations.*
5. ***Customizable Settings**: Users can adjust weather units and set alerts for severe weather.*
6. ***Live Map Streaming**: Shows a live map that updates with weather changes in real-time.*

Further details are available in Section 3.

2.3 User Classes and Characteristics

The intended users of WebCast include individuals of all skill levels, from students to professionals. No special expertise or experience is required. Users need only basic computing skills and familiarity with navigating web pages to access weather information and use the app's features effectively.

2.4 Operating Environment

WebCast will operate as a web-based application, accessible through modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. It is designed to function on both desktop and mobile devices, requiring an internet connection for real-time weather updates. The application is built using HTML, CSS, and JavaScript, ensuring cross-platform compatibility. WebCast will be compatible with major operating systems like Windows, macOS, and Linux for desktop users, as well as iOS and Android for mobile users. It will not require any additional plugins or software to run but will rely on standard web technologies, ensuring easy access and smooth performance across supported devices and operating systems. The app will also interface with weather data APIs for retrieving forecast information.

2.5 Design and Implementation Constraints

The design and implementation of WebCast are subject to several constraints. First, the app must adhere to web standards, requiring the use of HTML, CSS, and JavaScript for compatibility with modern browsers. It must support both desktop and mobile platforms, ensuring responsive design for varying screen sizes. The weather data will be retrieved via third-party APIs, meaning the app depends on the availability and reliability of external services. Security protocols must be in place for secure communication between the client and server, especially for location tracking and user

preferences. The app must operate within memory and performance limitations of typical web browsers, ensuring smooth functionality even on lower-spec devices. Additionally, WebCast must comply with any relevant data privacy regulations, particularly in handling user location and notification data. The development team will follow standard coding practices and ensure that the app is easily maintainable, with clear documentation for future updates.

2.6 User Documentation

The user documentation for WebCast will include the following components:

1. *User Manual: A comprehensive guide detailing the app's features, functions, and how to use them effectively.*
2. *Online Help: Context-sensitive help integrated within the app, providing users with instant assistance.*
3. *Tutorials: Step-by-step guides to help users get started with the app and explore its key features.*
4. *FAQ Section: A compilation of frequently asked questions to resolve common issues.*
5. *Documentation will be provided in digital formats, including HTML (for online access) and PDF (for offline use), ensuring easy access across devices.*

2.7 Assumptions and Dependencies

Assumptions for WebCast include reliable access to third-party weather data APIs, as the app depends on these for real-time updates. It is assumed that users will have stable internet connections for seamless functionality. The app's performance is also assumed to be optimal on standard web browsers and devices.

Dependencies include external services such as weather APIs, which may affect data accuracy or availability. Additionally, the app's functionality depends on the proper operation of location services in user devices. Changes in browser standards, security policies, or API terms could impact the app's performance and functionality.

3. System Features

Following are the functionalities of the WebCast application that are must for the system to work properly, to fulfil its purpose, and provide a desirable output.

3.1 Real Time Weather Display

Real-time weather display provides instant weather updates, including temperature, humidity, wind speed, and conditions, based on user location.

3.1.1 Description and Priority

This feature provides real-time weather updates for the user's current location. It is of **High priority** as it is the core functionality of the application.

3.1.2 Stimulus/Response Sequences

a. The user visits the website.
b. The system requests the user's location (with permission).
c. The system fetches and displays current weather data from an API.
d. If the user denies location access, the system prompts for manual location input.

3.1.3 Functional Requirements

a. **REQ-1:** The system must retrieve real-time weather data based on user location.
b. **REQ-2:** Users should be able to manually enter a location if geolocation is denied.
c. **REQ-3:** The UI must display temperature, humidity, wind speed, and weather conditions.
d. **REQ-4:** The system should update weather data every few minutes.

3.2 Weather Forecasts

Weather forecasts provide hourly and daily predictions, including temperature, precipitation, wind speed, and conditions, helping users plan ahead effectively.

3.2.1 Description and Priority

This feature provides hourly and daily weather forecasts. It is of **High priority** to help users plan ahead.

3.2.2 Stimulus/Response Sequences

a. The user selects a location.
b. The system fetches the weather forecast for the next few hours and days.
c. The data is displayed in an easy-to-read format (e.g., graph, table, or cards).

3.2.3 Functional Requirements

a. **REQ-1:** The system must fetch forecast data for at least 7 days.
b. **REQ-2:** Users should be able to toggle between hourly and daily forecasts.
c. **REQ-3:** The forecast should include temperature, precipitation, and wind speed.

3.3 Severe Weather Alerts

Severe weather alerts notify users of extreme conditions like storms, heavy rainfall, or heatwaves, ensuring timely warnings for safety and preparedness.

3.3.1 Description and Priority

This feature provides notifications for extreme weather conditions. It is of **medium priority** to ensure user safety.

3.3.2 Stimulus/Response Sequences

a. The system continuously monitors weather alerts from an API.
b. If a severe weather alert is detected, a notification is displayed to the user.

3.3.3 Functional Requirements

a. **REQ-1:** The system must fetch real-time severe weather alerts.
b. **REQ-2:** Alerts should be displayed prominently on the UI.
c. **REQ-3:** Users should receive notifications based on location settings.

4. External Interface Requirements

4.1 User Interfaces

The WebCast Weather Web Application will have a clean, responsive, and intuitive graphical user interface (GUI) designed using HTML, CSS, and JavaScript. The layout will include a navigation bar with sections for Current Weather, Forecast, Alerts, and Interactive Map. Standard buttons like Home, Refresh, and Help will be present on every screen. The UI will follow consistent styling with a color-coded weather representation. Keyboard shortcuts for quick navigation and error messages for invalid searches will be displayed clearly. Dynamic elements like live updates and animated weather icons enhance usability. The UI design will be documented separately.

4.2 Hardware Interfaces

*The **WebCast Weather Web Application** is designed to run on **desktop, laptop, tablet, and mobile devices** with modern web browsers. The application will support **touchscreen and non-touchscreen interfaces**, ensuring compatibility with **Windows, macOS, Android, and iOS** platforms. The system interacts with hardware components like **GPS modules** for location-based weather updates. Data exchange between the application and hardware will use **standard web communication protocols (HTTP/HTTPS)**. The software will efficiently handle **network connectivity variations** and support **offline caching** for previously fetched weather data. It requires **minimal CPU and RAM usage**, ensuring smooth performance on low-end devices.*

4.3 Software Interfaces

*The **WebCast Weather Web Application** will integrate with the **OpenWeather API (or similar)** to fetch real-time weather data, forecasts, and alerts in **JSON format**. It will run on **modern web browsers** (Chrome, Firefox, Edge, Safari) and use **JavaScript libraries** like **Fetch API** for API calls. The system will interact with **geolocation services** to detect user locations. No database is required, but local storage may be used for user preferences. The **API communication** will follow **HTTPS protocols** for security. The application will use **CSS frameworks** like Bootstrap for styling and adhere to standard web development best practices.*

4.4 Communications Interfaces

*The **WebCast Weather Web Application** will communicate with third-party **weather APIs** using **HTTP(S) requests** to fetch real-time weather data. The system will support **RESTful API integration** with JSON-formatted responses. Data transfer will be encrypted using **SSL/TLS** for security. The application will function in **modern web browsers** and require **internet connectivity** for live updates. Standard communication protocols like **TCP/IP and WebSockets** may be used for real-time updates. If email notifications are implemented, **SMTP or third-party services (e.g., SendGrid)** will be used. The system ensures **efficient data synchronization**, minimizing network bandwidth usage while maintaining timely weather updates.*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance should be able to support at least 10 simultaneous users. Data should be secured and backed up every day. Response time of the software is less than 5 seconds. Software should be operatable 24 hours and accessible in real time. Actions are performed very quickly. This software application will not stick and halt on any type of action.

5.2 Safety Requirements

WebCast will not damage or affect the other applications installed in the mobile phones and also not the performance of the mobile phone. The safety concerned with this software application that app should not be enabled when your phone is used by any other person and if you are crossing a road or may be ride a bicycle/bike/car and engage in other activity.

5.3 Security Requirements

This software app will be use when the user enters the login ID and password. App is used by only one user at a time. If the user enters wrong ID or password more than 3 times the message alert to administrator which will resend the password and login ID to your mobile number after verifying that you are the using this application. The software allows conventions which allow for familiar location for menus and other things.

5.4 Software Quality Attributes

The graphical user interface (GUI) design by usability. The app is user friendly and organized in this style that user can easy to navigate. There will be notifications about the latest weather forecast and any updates regarding application. To ensure reliability and correctness there will be zero forbearance for errors in the algorithms. For flexibility and adaptability if the user disconnects with the internet or having weak connectivity the user still able to use the application till the connection established.

6. Other Requirements

*The **WebCast Weather Web Application** must comply with **data privacy laws** regarding user location access. It should support **multiple languages** for international users. The application must ensure **API rate limits** are managed efficiently. Future scalability should allow integration with **additional weather data sources** and **AI-based forecasting models** for enhanced accuracy.*

Appendix A: Glossary

API	*Application Programming Interface*
REST	*Representational State Transfer*
DOM	*Document Object Model*
URL	*Uniform Resource Locater*
TC	*Test Case*

Appendix B: Issues List

a. ***API Selection: Final decision on the weather API provider (e.g., OpenWeather, WeatherAPI) is pending.***

b. ***Internationalization:*** *Languages to be supported are yet to be determined.*
c. ***Data Storage:*** *Decision on caching weather data for offline use is pending.*
d. ***Legal Compliance:*** *Privacy policy and terms of service need finalization.*
e. ***User Feedback Integration:*** *Mechanism for collecting and processing user feedback is yet to be defined.*