PES UNIVERSITY

Department of Computer Science and Engineering

UE21CS341A: Software Engineering

Software Requirements Specification

for

**A** **Weather Forecasting System**

**Version 1.0 approved**

**Prepared by Rahul Kalekar, R Sumedh, Raghavendra SB and Reehan Shaveez**

**PES UNIVERSITY**

**8-11-2023**

**Table of contents**

**Table of Contents...........................................................................................................................ii**

**Revision History.............................................................................................................................ii**

**1. Introduction..............................................................................................................................1**

1.1 Purpose..............................................................................................................................................1

1.2 Intended Audience and Reading Suggestions...................................................................................1

1.3 Product Scope...................................................................................................................................2

1.4 References.........................................................................................................................................4

**2. Overall Description..................................................................................................................5**

2.1 Product Perspective...........................................................................................................................5

2.2 Product Functions.............................................................................................................................5

2.3 User Classes and Characteristics......................................................................................................5

2.4 Operating Environment.....................................................................................................................7

2.5 Design and Implementation Constraints...........................................................................................8

2.6 Assumptions and Dependencies.......................................................................................................9

**3. External Interface Requirements...........................................................................................9**

3.1 User Interfaces..................................................................................................................................9

3.2 Software Interfaces...........................................................................................................................10

3.3 Communications Interfaces..............................................................................................................11

**4. Analysis Models**

**5. System Features.......................................................................................................................14**

5.1 System Feature 1...............................................................................................................................14

5.2 System Feature 2 (and so on) ............................................................................................................15

**6. Other Nonfunctional Requirements.......................................................................................17**

6.1 Performance Requirements...............................................................................................................17

6.2 Safety Requirements.........................................................................................................................18

6.3 Security Requirements......................................................................................................................19

6.4 Software Quality Attributes..............................................................................................................22

6.5 Business Rules..................................................................................................................................25

**7. Other Requirements................................................................................................................29**

**Appendix A: Glossary....................................................................................................................31**

**Appendix B: Field Layouts...........................................................................................................32**

**Appendix C: Requirement Traceability matrix..........................................................................35**

Introduction

**Purpose**:

The purpose of the product is to assist in predicting weather conditions of a given area thereby helping in averting timely accidents, assist in adoption of better preventive measures for handling disasters and assist in various other fields where assessment and analysis of weather plays an important role.

**Intended Audience**:

**Users**: This document delves in a very precise manner into the various aspects of this weather forecasting software. The language is simple enough for users to familiarize themselves with the various functionalities of this product.

**Marketing staff**: they can gain an insight into the various functionalities of the product and market it accordingly.

**Developers:** Developers are provided a detailed insight into the various functionalities of the product and they can utilize the same to make future enhancements to the product.

The rest of the product contains a detailed overview about the functions of the product, the intended users etc.

**Product Scope**

The software that is being developed is a weather forecasting system. The purpose of this system would be to predict the weather, given a particular area. The predictions are not only confined to the present but also to the past and the near future

Key Objectives and Goals:

1. **Accurate weather predictions**: Make accurate and real time analysis of current weather conditions.
2. **User Friendly Interface**: Users will be able to make use of the software very easily with minimum difficulty. They can easily access the software on their mobile devices as well as laptops
3. **Timely Updates**: Weather information should be updated in real-time or at frequent intervals to reflect the current weather conditions and any changes that may occur.
4. **Customized alerts**: Users should be warned of severe weather conditions if there are any. This would prevent accidents.

Alignment with corporate goals:

1. **Enhancing User Experience**: By offering accurate and reliable weather predictions, the software can improve user satisfaction and engagement. This, in turn, can drive increased user retention and loyalty.
2. **Supporting Decision-Making**: Providing detailed weather forecasts can aid individuals and businesses in making better decisions, such as planning outdoor events, optimizing travel routes, or managing agricultural operations. This can contribute to overall efficiency and effectiveness.
3. **Competitive Advantage**: A high-quality weather forecasting system can differentiate a business in sectors like travel, agriculture, logistics, and outdoor event planning. It can attract and retain users, thereby improving market position and competitiveness.
4. **Data Monetization**: Weather data can have commercial value. Businesses can explore opportunities to monetize weather data by offering premium services or data insights to industries that rely on accurate weather information.

**References:**

Youtube Videos related to prediction weather using apis and frontend development

Rapid Api Website to explore weather api’s

**Overall Description**

**Product Perspective**

This product is a follow-on product of the software that government agencies such as The Indian Meteorological Department, National Centers for Environmental Prediction, NASA make use of.

**Product Functions**

1. Predict the weather conditions of a particular area. Provide information to users in terms of temperature, humidity etc.
2. Provide alerts to users in case of severe weather conditions

**User characteristics**

The various user classes that will use this product are:

1. **General Public Users**:

Characteristics: These users are the general public who access weather forecasts for daily planning and information.

Requirements: They require user-friendly interfaces, easy-to-understand weather information, and location-based forecasts. Timely and accurate updates are essential.

1. **Outdoor Event Planners**:

Characteristics: Event planners, organizers, and individuals planning outdoor activities and events.

Requirements: Detailed weather forecasts, including precipitation, wind speed, and UV index, are vital. They may need long-range forecasts to plan events well in advance.

1. **Travelers**:

Characteristics: Tourists, business travelers, and commuters who need weather forecasts for travel planning.

Requirements: Access to real-time weather information, airport-specific forecasts, and travel safety alerts for potential disruptions.

1. **Farmers and Agriculturists**:

Characteristics: Agriculture professionals and farmers who rely on weather data for crop management.

Requirements: Specialized agricultural weather forecasts, including soil moisture, frost predictions, and planting and harvesting recommendations.

1. **Emergency Services**:

Characteristics: First responders, emergency management teams, and public safety agencies.

Requirements: Timely severe weather alerts, flood predictions, and disaster preparedness information.

1. **Media and News Outlets**:

Characteristics: Media professionals and news agencies providing weather reports to the public.

Requirements: Reliable and up-to-date weather information, visualization tools, and graphics for broadcasting.

1. **Educational and Research Institutions**:

Characteristics: Schools, colleges, and universities using weather data for educational purposes.

Requirements: Access to educational resources, weather data APIs, and student-friendly interfaces.

**Operating Environment**

Can be operated on any operating system on a laptop. Can be operated on android systems in mobile phones.

**Design and Implementation Constraints**

1. **Data Sources and Quality**:

**Limitation**: The availability and quality of weather data from various sources can be a limitation. Inaccurate or incomplete data can impact the accuracy of forecasts.

1. **Computational Resources:**

**Limitation**: Weather forecasting applications often require significant computational resources, including powerful hardware and cloud infrastructure.

1. **Interoperability:**

**Limitation**: The system may need to interface with various applications, platforms, and devices, requiring interoperability considerations.

1. **Scalability and Performance:**

**Limitation**: Ensuring the system's performance under increased load and scaling can be challenging.

**Assumptions**

One of the most important assumptions that is made while creating this software is that data related to weather conditions for a particular area is available and ready at hand. This plays an important role as the availability of data can make or break the creation of the project.

The second assumption is that the application/product if developed can be easily integrated in any operating environment. As newer versions of operating systems are being released, in the worst-case scenario the application might not be supported in one of these versions.

**External User Interfaces**

**User Interfaces**

Here is an overview of characteristics that can be considered are:

1. **User Interface Purpose:** The web-based dashboard servers as the primary user interface for accessing weather data. Its purpose is to provide users with weather information, forecasts, and historical data in real time.
2. **Components:** It includes the components like widgets, maps, search bars for selection of location and other customization options for displaying specific data requested by the user.
3. **Data Presentation:** It should present data through graphs, text, charts, tables, offering options to customize to display the preferred data sources.
4. **User Controls:** The interface should offer a variety of controls, such as providing the option of searching from your own location, providing the option to visualize graphs etc.

This interface may require a more detailed use interface for the design, layout, interaction patterns.

It should include wireframes, user flows, design mockups and usability guidelines to ensure a consistent and user-friendly experience.

**Software Interfaces**

The possible connections and their purposes are:

1. **Databases:** Weather monitoring systems often rely on DBMS to store and manage large volumes of historical and real-time weather data. Common databases include MySQL or NoSQL like MongoDB. These store various types of data, such as temperature, humidity, precipitation, wind speed, and historical weather records. They help and facilitate data retrieval and analysis.
2. **Operating Systems:** There are two types of OS, Server OS and Client OS. The system’s servers, where data is processed, run on operating systems like Linux or Windows server. Whereas for clients, either mobile apps / webpages for weather monitoring may be developed for specific operating systems. OS provides the foundation for running the system and supporting its various components.
3. **Tools and Libraries:**
4. **Communication Protocols:** These enable data exchange between the system and external components.

These tools and libraries enhance data visualization, analysis, and communication with external services or clients.

1. **Integrated Commercial Components:**
2. **Weather Data Providers:** This system integrates with commercial weather data providers to obtain real-time and forecast data.
3. **Advertising** **Services:** If the system includes advertisements, it might integrate with ad networks.

These components provide external data, monetization, options and services.

1. **Data Items and Messages:**
2. **Incoming Data Items:** Weather data (such as temperature, humidity, wind speed ), Location Data (user-selected or GPS coordinates.), User preferences (notification settings)
3. **Outgoing Data Items:** Weather Forecasts and Current Conditions, Alerts and Notifications, User-Generated Data.

Incoming data items are collected from various sources and are processed and used to generate outgoing data items. These connections and components are critical for proper functioning of a weather monitoring system, allowing it to collect, process, and deliver accurate and timely information to users.

**Communication Interfaces**

The requirements and considerations associated with communication functions of the product are:

1. **Data Transfer Rates and Synchronization:**
2. **Real-time data transfer:** real-time weather data from sensors to system for immediate analysis and processing. We should ensure low latency and data transfer rates to support real-time monitoring and timely decision-making.
3. **Periodic data synchronization:** Its purpose is to synchronize historical data between databases and analysis tools. We should periodically update (define periodic synchronization intervals) databases with the latest processed data for forecasting and analysis.

2. **Communication Protocols and Standards:**

1. **Internet Protocols (TCP/IP):** It enables communication between different components over the internet. But the requirements are all the components must support TCP/IP for seamless communication

2. **HTTP/HTTPS:** It facilitates communication between web browsers and the system’s web server. It should support Web interfaces for data visualization and user interaction must use HTTPS to ensure data security during transmission.

3. **Web Browser:**

**Requirements:** Browsing websites, rendering HTML/CSS content, executing JavaScript,

handling cookies.

**Communication Standards:** HTTP/HTTPS for web communication.

**Security:** Implement security mechanisms like HTTPS to ensure encrypted communication

between the browser and web server.

4. **Electronic Forms:**

**Requirements:** Creating, transmitting and storing electronics forms data.

**Message Formatting:** Use standardized formats like XML or JSON to structure form data

**Communication Standards:** HTTP/HTTPS for form submissions to web servers.

5. **Email Communication:**

**Requirements:** Sending and receiving emails, attaching files, managing email threads.

**Message Formatting:** Use standard email message formats like MIME (Multipurpose Internet Mail Extensions) for multimedia attachments.

**Communication Standards:** SMTP (Simple Mail Transfer Protocol) for sending emails, IMAP (Internet Message Access Protocol) and POP3 (Post Office Protocol) for receiving emails.

**Security:** Use SSL/TLS encryption for secure email communication to prevent interception and unauthorized access.

It is essential to adhere to industry standards and follow the best practices for security. Consider the specific use case to determine the most suitable communication protocols and technologies to use.

**Analysis Models**

*<TO DO>*

**System Features**

**Real-time Weather Data Retrieval**

Description and priority:

This system feature enables users to request real-time weather data for specified

locations. It is of high priority as it forms the core functionality of the weather monitoring

system.

Stimulus/Response Sequences:

* User inputs location and requests weather data.
* System queries the weather database based on the provided location.
* System retrieves real-time weather data (temperature, humidity, wind speed, etc.).
* System displays the weather data to the user.

Functional Requirements:

* REQ-1: The system must provide an input field for users to enter the location (city, zip code, coordinates, etc.).
* REQ-2: The system must have access to a reliable weather database to fetch real-time weather information.
* REQ-3: The system must display retrieved weather data, including temperature, humidity, wind speed, and weather conditions.
* REQ-4: The system must update weather data at regular intervals to ensure accuracy.
* REQ-5: The system must handle cases where the entered location is invalid or not found in the database and provide appropriate error messages to the user.

**Weather Forecasting**

Description and Priority:

This feature enables users to view weather forecasts for a specific location over a defined

period. It is of medium priority as it enhances the user experience but is not as critical as

real-time data retrieval.

Stimulus/Response Sequences:

* User selects a specific location and time frame for the weather forecast.
* System queries the forecasting algorithm based on the selected location and time frame.
* System retrieves weather forecast data (daily or hourly predictions).
* System displays the weather forecast data to the user.

Functional Requirements:

* REQ-1: The system must utilize a reliable forecasting algorithm to generate accurate weather predictions.
* REQ-2: The system must display the weather forecast data, including temperature trends, precipitation chances, and wind forecasts.
* REQ-3: The system must allow users to compare forecasted data for different time frames and locations.
* REQ-4: The system must handle cases where the forecast data is not available or cannot be generated and provide appropriate notifications to the user.

**Other Non Functional Requirements :**

**Performance Requirements**

* The weather forecast detection model should be able to generate a forecast for a given location within 100 milliseconds, 95% of the time.
* The weather forecast detection model should be able to generate forecasts for multiple locations without any significant performance degradation.
* The weather forecast detection model should be able to handle a variety of different weather conditions, including thunderstorms, hurricanes, and blizzards.

The overall performance requirements for the web application are designed to ensure that it is responsive and scalable. The response time requirement of 1 second is based on the expectation that users will become frustrated if they have to wait longer than that for a response. The ability to handle 1000 concurrent users is based on the expectation that the web application will be popular and will need to be able to handle a large number of users at the same time. The scalability requirement is designed to ensure that the web application can be scaled up to handle more users and traffic in the future.

The specific performance requirements for weather forecast detection are designed to ensure that the web application can provide users with accurate and timely weather forecasts. The response time requirement of 100 milliseconds is based on the expectation that users will need to be able to get weather forecasts quickly, especially in emergency situations. The ability to generate forecasts for multiple locations simultaneously is based on the expectation that users will often be interested in getting forecasts for multiple locations, such as their home, work, and travel destinations. The ability to handle a variety of different weather conditions is designed to ensure that the web application can provide users with accurate forecasts even in extreme weather conditions**.**

**Safety Requirements**

The following are some possible losses, damages, or harms that could result from the use of a weather forecast detection web application:

Users may make decisions that could put themselves or others in danger if they rely on inaccurate or outdated weather forecasts. For example, a user may decide to go hiking in a thunderstorm if they are not aware that a severe weather warning has been issued.

Users may experience financial losses if they rely on inaccurate weather forecasts to make business decisions. For example, a farmer may decide to plant a crop that is not suited for the expected weather conditions, which could lead to crop failure.

Users may experience emotional distress if they are not prepared for severe weather conditions. For example, a user may become anxious or scared if they are caught in a thunderstorm without adequate shelter.

Safeguards and Actions

The following are some safeguards and actions that can be taken to mitigate the risks of loss, damage, or harm from using a weather forecast detection web application:

The application should prominently display a disclaimer that states that the forecasts are not guaranteed to be accurate. This will help users to understand that they should use the forecasts at their own risk.

The application should provide users with the ability to customize the frequency and type of alerts they receive. This will allow users to receive alerts for the weather conditions that are most important to them.

The application should be developed and maintained in accordance with best practices for software development. This will help to reduce the risk of software defects that could lead to inaccurate forecasts.

**External Policies and Regulations**

The following are some external policies and regulations that state safety issues that affect the design or use of weather forecast detection web applications:

The National Oceanic and Atmospheric Administration (NOAA) has a policy that requires weather forecast products to be disseminated in a way that is accurate, timely, and reliable. This policy is designed to protect the public from harm due to inaccurate or outdated weather forecasts.

The Federal Emergency Management Agency (FEMA) has a regulation that requires states and local governments to develop and implement hazard mitigation plans. These plans must include a component for addressing the risks of severe weather events.

Safety Certifications

There are no specific safety certifications that are required for weather forecast detection web applications. However, developers may want to consider obtaining certifications such as ISO 27001 (Information Security Management System) or ISO 9001 (Quality Management System) to demonstrate that their application has been developed and is maintained in accordance with best practices.

**Security Requirements**

**Security and Privacy Issues**

The following are some security and privacy issues that should be considered for a weather forecast detection web application:

Unauthorized access to user data. The application should take steps to protect user data from unauthorized access, such as using encryption and strong authentication mechanisms.

Data breaches. The application should have a plan in place to respond to data breaches, including notifying affected users and taking steps to mitigate the damage.

Denial of service attacks. The application should be designed to be resilient to denial of service attacks, which can prevent users from accessing the application.

Cross-site scripting (XSS) attacks. The application should be protected from XSS attacks, which can allow attackers to inject malicious code into the application and execute it on the client side.

User Identity Authentication Requirements

The following are some user identity authentication requirements that should be considered for a weather forecast detection web application:

Users should be required to create an account and authenticate themselves before accessing the application. This will help to prevent unauthorized access to the application.

The application should use strong authentication mechanisms, such as two-factor authentication. This will help to protect user accounts from being compromised.

The application should allow users to reset their passwords if they forget them. This will help to ensure that users can always access their accounts.

**External Policies and Regulations**

The following are some external policies and regulations that contain security issues that affect the design or use of weather forecast detection web applications:

The General Data Protection Regulation (GDPR) is a regulation in the European Union that requires organizations to protect the personal data of EU residents. This regulation may apply to weather forecast detection web applications that collect or process personal data from EU residents.

The California Consumer Privacy Act (CCPA) is a law in the state of California that gives consumers the right to know what personal data is being collected about them, the right to request that their personal data be deleted, and the right to opt-out of the sale of their personal data. This law may apply to weather forecast detection web applications that collect or process personal data from California residents.

**Security and Privacy Certifications**

The following are some security and privacy certifications that may be relevant for weather forecast detection web applications:

ISO/IEC 27001 is an international standard for information security management. This certification demonstrates that an organization has implemented a comprehensive information security management system.

SOC 2 is a type of service organization control (SOC) report that focuses on security, availability, processing integrity, confidentiality, and privacy. This certification demonstrates that an organization has implemented and maintains effective controls to protect customer data.

PCI DSS is a payment card industry data security standard that establishes requirements for organizations that store, process, or transmit credit card data. This certification may be relevant for weather forecast detection web applications that accept credit card payments.

**Software Quality Attributes**

Adaptability: The application should be adaptable to changes in the weather forecast detection and to changes in the user interface. This can be achieved by using a modular architecture and by designing the application to be easily extensible.

Availability: The application should be available to users 99.9% of the time. This can be achieved by using a high-availability hosting infrastructure and by implementing load balancing to distribute traffic across multiple servers.

Correctness: The application should generate accurate weather forecasts for a variety of different locations and weather conditions. This can be achieved by using multiple sources of data to train the weather forecast detection model and by validating the model's predictions against historical weather data.

Flexibility: The application should be flexible enough to accommodate a variety of different user needs. This can be achieved by providing a variety of different customization options and by making the application easy to integrate with other applications.

Interoperability: The application should be interoperable with other weather-related applications and with other systems that the user may be using. This can be achieved by using standard data formats and by providing well-documented APIs.

Maintainability: The application should be easy to maintain and update. This can be achieved by using a modular architecture, by writing well-documented code, and by using automated testing to ensure that the code remains correct after changes are made.

Portability: The application should be portable to different operating systems and hardware platforms. This can be achieved by using a cross-platform programming language and by avoiding platform-specific APIs and libraries.

Reliability: The application should be reliable and should not crash or lose data. This can be achieved by implementing error handling and recovery mechanisms and by testing the application thoroughly.

Reusability: The application should contain reusable components that can be used in other weather-related applications. This can be achieved by designing the application in a modular way with well-defined interfaces between the modules.

Robustness: The application should be robust to unexpected inputs and errors. This can be achieved by validating user input and by implementing error handling and recovery mechanisms.

Testability: The application should be easy to test. This can be achieved by designing the application in a modular way and by writing unit tests for all of the code.

Usability: The application should be easy to use and understand. This can be achieved by designing a user-friendly interface and by providing clear and concise instructions.

**Quantitative Requirements**

The following are some quantitative requirements for the software quality attributes listed above:

* Availability: The application should be available to users 99.9% of the time.
* Correctness: The application should generate accurate weather forecasts for 95% of locations and weather conditions.
* Flexibility: The application should be able to accommodate 90% of user needs.
* Interoperability: The application should be interoperable with 80% of other weather-related applications.
* Maintainability: The application should have a maintainability index of at least 80%.
* Portability: The application should be portable to 90% of operating systems and hardware platforms.
* Reliability: The application should not crash or lose data more than 0.1% of the time.
* Reusability: The application should contain at least 50% reusable components.
* Robustness: The application should be able to handle 95% of unexpected inputs and errors without crashing or losing data.
* Testability: The application should have a test coverage of at least 90%.
* Usability: The application should have a usability score of at least 85%, as measured by the System Usability Scale (SUS).

**Relative Preferences**

The relative preferences for the software quality attributes listed above are as follows:

Accuracy: This is the most important attribute for a weather forecast detection web application. Users rely on the accuracy of the forecasts to make important decisions about their safety and well-being.

Reliability: This attribute is also very important, as users need to be able to rely on the application to provide accurate forecasts.

Usability: The application should be easy to use and understand for everyone, regardless of their technical expertise.

Maintainability: The application should be easy to maintain and update so that it can keep up with the latest weather forecast detection technology.

Portability: The application should be portable to a variety of operating systems and hardware platforms so that it can be used by as many people as possible.

The other software quality attributes are also important, but they are not as critical as the accuracy, reliability, usability, maintainability, and portability attributes.

**Business Rules**

* User registration: Users must register for an account before they can use the application.
* User authentication: Users must authenticate themselves before they can access the application's features.
* Authorization: Users are only authorized to access the features of the application that are relevant to their role. For example, administrators may be authorized to view and edit all of the application's data, while users may only be authorized to view their own weather forecasts.
* Data access: Users are only granted access to the data that they need to perform their job function. For example, a user who is responsible for monitoring the weather in a specific region may only be granted access to the weather forecast data for that region.
* Data accuracy: The application's weather forecast data must be accurate and up-to-date.
* Data security: The application's weather forecast data must be protected from unauthorized access and modification.
* Audit trail: The application must maintain an audit trail of all user activity. This audit trail can be used to track who accessed the application, what data they accessed, and when they accessed it.

**Operating Principles**

The following are some operating principles for the weather forecast detection web application:

Accessibility: The application should be accessible to users of all abilities.

Reliability: The application should be available to users 24/7.

Security: The application should be secure from unauthorized access and modification.

Performance: The application should respond to user requests quickly.

Scalability: The application should be able to handle a large number of users and requests without any significant performance degradation.

**How Business Rules Imply Functional Requirements?**

The following are some examples of how business rules imply functional requirements:

The business rule "Users must register for an account before they can use the application" implies the following functional requirement: "The application must provide a user registration feature."

The business rule "Users are only authorized to access the features of the application that are relevant to their role" implies the following functional requirement: "The application must implement an authorization mechanism."

The business rule "Data must be protected from unauthorized access and modification" implies the following functional requirement: "The application must implement authentication and authorization mechanisms."

The business rule "The application must maintain an audit trail of all user activity" implies the following functional requirement: "The application must implement an audit logging mechanism."

**Domain requirement**

The following are some domain requirements for a weather forecast detection web application:

The application should be able to generate accurate weather forecasts for a variety of different locations and weather conditions.

The application should be able to detect severe weather events, such as tornadoes, hurricanes, and floods.

The application should be able to provide users with alerts about severe weather events.

The application should be able to integrate with other weather-related systems, such as weather stations and radar systems.

The application should be able to provide users with access to historical weather data.

**Specific Domain Requirements**

The following are some specific domain requirements for the weather forecast detection web application:

The application should be able to generate weather forecasts for the following locations:

* Cities and towns
* Airports
* Zip codes
* Latitude and longitude coordinates

The application should be able to generate weather forecasts for the following weather conditions:

* Temperature
* Humidity
* Wind speed and direction
* Precipitation
* Cloud cover
* Air quality

The application should be able to detect the following severe weather events:

Tornadoes

* Hurricanes
* Floods
* Winter storms
* Thunderstorms

The application should be able to provide users with the following alerts about severe weather events:

* Email alerts
* SMS alerts
* Push notifications

The application should be able to integrate with the following weather-related systems:

* Weather stations
* Radar systems
* Satellite systems

The application should be able to provide users with access to historical weather data for the following time periods:

* 1 day
* 1 week
* 1 month
* 1 year
* 10 years

**Other Requirement**

Database Requirements

The weather forecast detection web application will require a database to store the following data:

* User data
* Weather forecast data
* Severe weather event data
* Historical weather data
* The database should be scalable and reliable, and it should be able to handle a large number of concurrent users and requests.

Internationalization Requirements

The weather forecast detection web application should be internationalized so that it can be used by users around the world. This means that the application should be able to display text and dates in different languages and formats. The application should also be able to handle different time zones.

Legal Requirements

The weather forecast detection web application must comply with all applicable laws and regulations. This includes laws and regulations related to data privacy and security.

Reuse Objectives

The following components of the weather forecast detection web application should be reusable:

The weather forecast detection model

The user authentication and authorization system

The database schema

By reusing these components, developers can save time and effort when developing other weather-related applications.

Other Pertinent Requirements

The following are some other pertinent requirements for the weather forecast detection web application:

* The application should be accessible to users of all abilities.
* The application should be easy to use and understand.
* The application should be visually appealing.
* The application should be performant and scalable.
* The application should be secure from unauthorized access and modification.

**Appendix A: Glossary**

**Acronyms and Abbreviations**

API: Application Programming Interface

CPU: Central Processing Unit

DBMS: Database Management System

GUI: Graphical User Interface

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

IP: Internet Protocol

ISP: Internet Service Provider

RAM: Random Access Memory

SQL: Structured Query Language

SRS: Software Requirements Specification

TCP: Transmission Control Protocol

UI: User Interface

**Terms**

Accuracy: The degree to which the weather forecast detection web application's predictions match the actual weather conditions.

Availability: The percentage of time that the weather forecast detection web application is available to users.

Database: A collection of data that is organized in a way that makes it easy to retrieve and manage.

Functional Requirement: A requirement that specifies what the weather forecast detection web application must do.

Non-Functional Requirement: A requirement that specifies how the weather forecast detection web application must perform, such as accuracy, availability, performance, and scalability.

Performance: The speed at which the weather forecast detection web application can respond to user requests.

Portability: The ability of the weather forecast detection web application to run on different operating systems and hardware platforms.

Reliability: The ability of the weather forecast detection web application to perform its intended function without failing.

Robustness: The ability of the weather forecast detection web application to continue operating even when there are unexpected errors or failures.

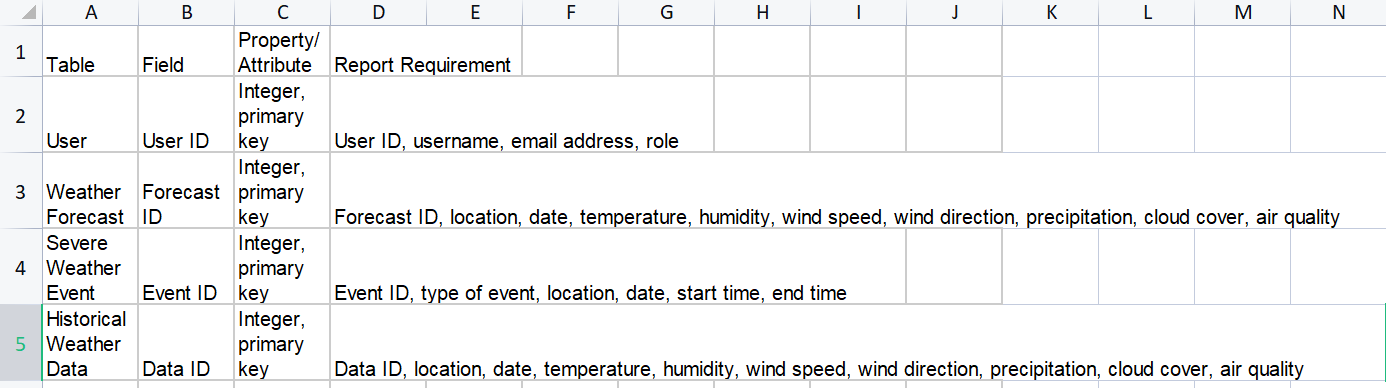
Scalability: The ability of the weather forecast detection web application to handle a large number of users and requests without any significant performance degradation.

Security: The ability of the weather forecast detection web application to protect user data and functionality from unauthorized access and modification.

Usability: The ease with which users can learn and use the weather forecast detection web application

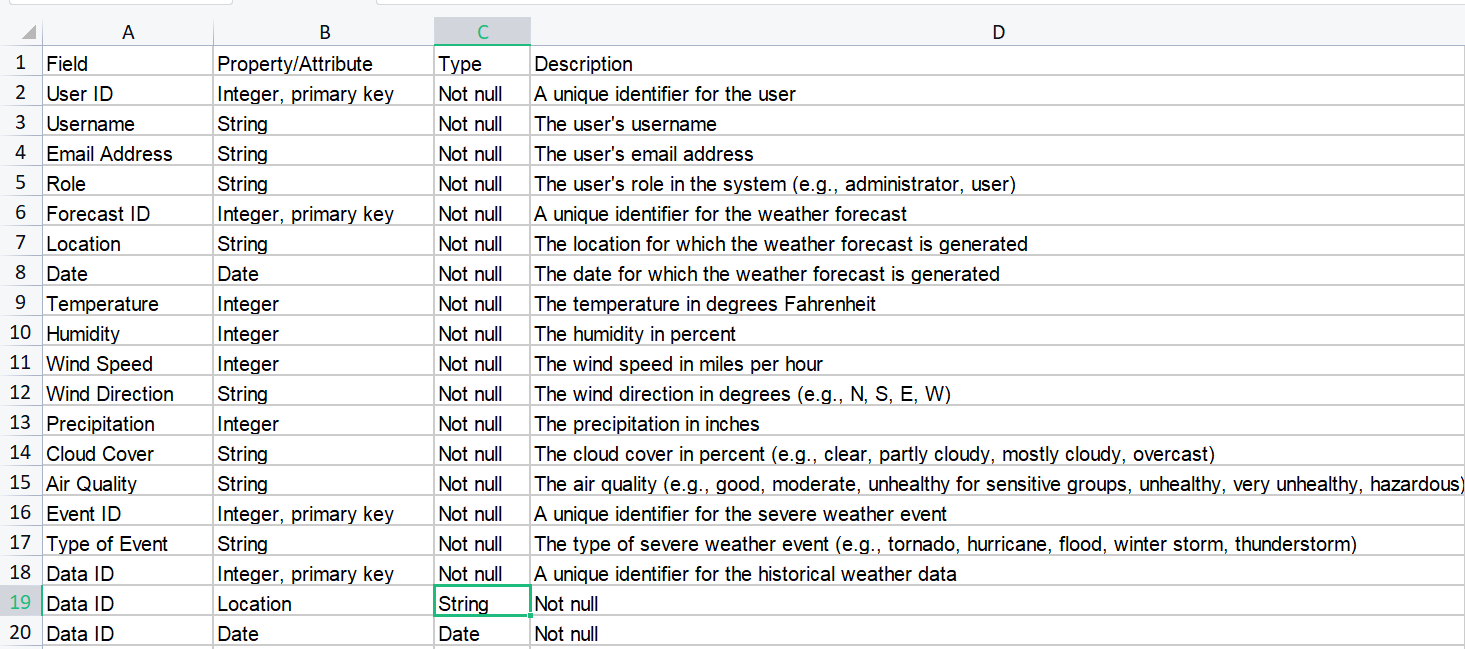
**Appendix B: Field Layouts**

**The following table shows the field layouts for the weather forecast detection web application:**



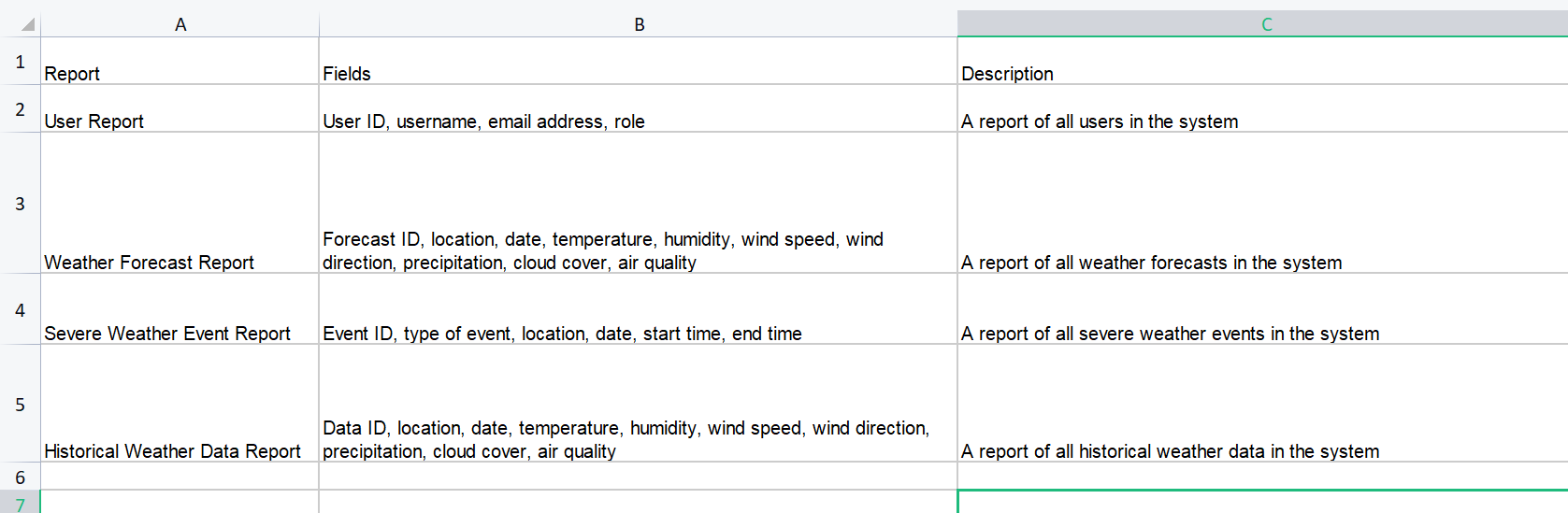
**Field Properties and Attributes**

**The following table shows the properties and attributes for the fields in the weather forecast detection web application:**



**Report Requirements**

**The following table shows the report requirements for the weather forecast detection web application:**



**Appendix C: Requirement Traceability Matrix**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S no** | **Requirement**  **ID** | **Brief Description**  **of Requirement** | **Architecture**  **Reference** | **Design**  **Reference** | **Code File**  **Reference** | **Test**  **Case ID** | **System**  **Test**  **Case ID** |
| 1 | 1 | Weather api for fetching weather for a given location | rapidapi.com/apininjas/api/weather-by-api-ninjas/ | rapidapi.com/apininjas/api/weather-by-api-ninjas/ | rapidapi.com/apininjas/api/weather-by-api-ninjas/ | 1 | 1 |
| 2 | 2 | Search bar for giving location as an input | Bootstrap website | Bootstrap website | https://getbootstrap.com/ | 2 | 2 |
| 3 | 3 | Alert Button for displaying alerts | Bootstrap Website | Bootstrap Website | https://getbootstrap.com/ | 3 | 3 |