Software Requirement Specification

For

Weather Data Analysis Application

Prepared by

Rangan Satpathi   12021002016034   rangansatpathi17@gmail.com

Ritankar Das   12021002016046   ritankar25@gmail.com

Instructor: Prof. Bipasha Mahato

Prof. Deepsubhra Guha Roy

Course: Database Management System

Contents

Introduction

Weather data is very crucial in every aspect of human daily life. It plays an important role in many sectors such as agriculture, tourism, government planning, industry and so on. Weather has a variety of parameters like temperature, pressure, humidity and wind speed. So a weather data analysis app is very useful for our everyday life.

Document Purpose:

The Weather Data Analysis Application is a software solution designed to collect, process, analyze, and visualize weather-related information. This document outlines the specifications, functionalities, and requirements of the application, which aims to provide accurate and comprehensive weather insights to users. The primary purpose of this SRS is to define the scope of the project, describe its objectives, and outline the features and constraints.

Product Scope:

The Weather Data Analysis Application will empower users, including meteorologists, researchers, and weather enthusiasts, to access, manipulate, and interpret weather data for various purposes, such as forecasting, research, and decision-making. The application will encompass features for gathering real-time and historical weather data, performing in-depth data analysis, generating reports, and creating interactive visualizations.

Document Conventions:

Bold Text: Used for section headings.

Italic Text: Used for placeholders and variable names.

Monospaced Text: Used for code snippets, file names, and technical terms.

References And Acknowledgments:

[Project Proposal Document]: The initial project proposal document, dated [Insert Date], serves as a reference for the project's goals and objectives.

[System Architecture Document]: The System Architecture Document outlines the high-level system architecture and design, providing additional context for this SRS.

Overview:

This document will detail the following aspects of the Weather Data Analysis Application:

Functional Requirements: Describes the application's features and how they interact with users.

Non-functional Requirements: Specifies the performance, usability, security, and other quality attributes.

External Interfaces: Outlines the external systems, APIs, and databases that the application interacts with.

Constraints: Identifies any limitations or restrictions on the project.

Assumptions and Dependencies: Lists the assumptions made during the project and the external factors the application depends on.

Audience:

This document is intended for various stakeholders, including:

Development Team: To understand the technical requirements and constraints.

Project Managers: To ensure that the project aligns with the initial proposal.

Quality Assurance Team: To develop test cases and ensure the application meets the specified requirements.

Users: To gain an understanding of the application's capabilities and limitations.

Specific Requirements

In this section of the Software Requirements Specification (SRS) document for the Weather Data Analysis Application, we will outline the specific functional and non-functional requirements that need to be met for the successful development of the application.

External Interface Requirements:

In this section of the Software Requirements Specification (SRS) document for the Weather Data Analysis Application, we will outline the external interfaces that the application will interact with. These interfaces include external systems, APIs, databases, and hardware components that are integral to the functionality of the application.

User Interface:

The Weather Data Analysis Application will have a user-friendly graphical user interface (GUI) accessible through web browsers on various devices. The GUI will provide users with the ability to interact with the application's features, including data retrieval, analysis, visualization, and reporting.

Weather data provider:

The application will interface with one or more external weather data providers to retrieve current weather conditions, forecasts, and historical weather data. Examples of such providers include:

OpenWeatherMap API: Provides access to global weather data, including current conditions and forecasts.

Weather.com API: Provides weather data and forecasts for various locations.

Databases:

The Weather Data Analysis Application may interact with one or more databases to store user profiles, preferences, historical data, and generated reports. The choice of database management systems (DBMS) will depend on project requirements and constraints. Commonly used DBMS include:

MySQL: An open-source relational database management system.

Hardware Interfaces:

The application does not have specific hardware interfaces as it primarily operates in a web-based environment. However, it may utilize standard hardware components such as servers, network infrastructure, and storage devices to host and maintain the application.

Software Interfaces:

The Weather Data Analysis Application may integrate with other software systems and libraries to enhance its functionality. These software interfaces may include:

Operating Systems: The application should be compatible with various operating systems, including Windows, Linux, and macOS.

Web Browsers: The GUI should function correctly on popular web browsers such as Chrome, Firefox, Safari, and Edge.

Programming Languages and Frameworks: The application may be built using programming languages such as JavaScript, Python, and frameworks like Django or Flask.

Third-party Libraries: The application may use third-party libraries and modules for data analysis, visualization, and reporting.

Functional Requirements:

Functional requirements describe the specific features and interactions that the Weather Data Analysis Application should support.

User Registration and Authentication:

The application shall allow users to create an account with a unique username and password. Users shall be able to log in using their credentials. Passwords must be securely stored using encryption techniques. Users shall have the option to reset their password if forgotten.

Data connection and retrieval:

The application shall provide access to real-time weather data, including temperature, humidity, wind speed, and precipitation, for a user-defined location. Users shall be able to specify the date and time range for historical weather data retrieval. The application shall connect to external weather data sources and APIs to fetch data. Data shall be presented in a user-friendly format, such as charts, graphs, and tables.

Data Analysis and Visualization:

Users shall have the ability to perform various data analysis operations, including statistical analysis, trend analysis, and anomaly detection. The application shall provide tools for users to create interactive visualizations, such as time series plots, heatmaps, and spatial maps. Users shall be able to customize visualization parameters, including color schemes and data overlays.

Reporting:

The application shall allow users to generate reports based on their data analysis and visualizations. Reports shall be exportable in commonly used formats, such as PDF or CSV. Users shall have the option to save and manage their reports within the application.

User Profile and Preferences:

Users shall be able to update their profile information, including contact details and notification preferences. The application shall save user preferences for data presentation, default locations, and analysis settings.

Administrative Features:

Administrators shall have access to manage user accounts, permissions, and system settings.

Administrators shall be able to monitor system performance and troubleshoot issues.

Non-Functional Requirements:

Non-functional requirements define the qualities of the Weather Data Analysis Application.

Performance:

The application shall provide real-time data with a response time of less than 2 seconds for data retrieval and visualization. The system shall support concurrent access by multiple users without performance degradation.

Security:

User data, including login credentials and personal information, shall be securely stored and transmitted using encryption techniques. The application shall implement user access control to ensure data privacy and security. The system shall comply with industry-standard security practices and regulations.

Usability:

The application shall have an intuitive and user-friendly graphical user interface (GUI) with responsive design for various devices and screen sizes. Users shall be provided with contextual help and documentation for using the application's features. The application shall support accessibility features for users with disabilities.

Scalability:

The system architecture shall be designed to scale horizontally to accommodate increased user loads. The application shall support the addition of new external data sources and APIs.

Compatibility:

The application shall be compatible with commonly used web browsers (e.g., Chrome, Firefox, Safari). The system shall be designed to work on multiple operating systems and devices.

Data Integrity:

Data imported into the application shall be validated for accuracy and consistency. The application shall log data changes and user actions for auditing purposes.

Data Flow Diagrams

A Data Flow Diagram (DFD) is a visual representation of the flow of data within a system or process. It illustrates how data is input, processed, stored, and output in a system. DFDs use symbols to represent processes, data stores, data flows, and external entities. Here, I'll provide a simplified DFD for a Weather Data Analysis Application to demonstrate its key components and data flows.

Level–0–DFD:
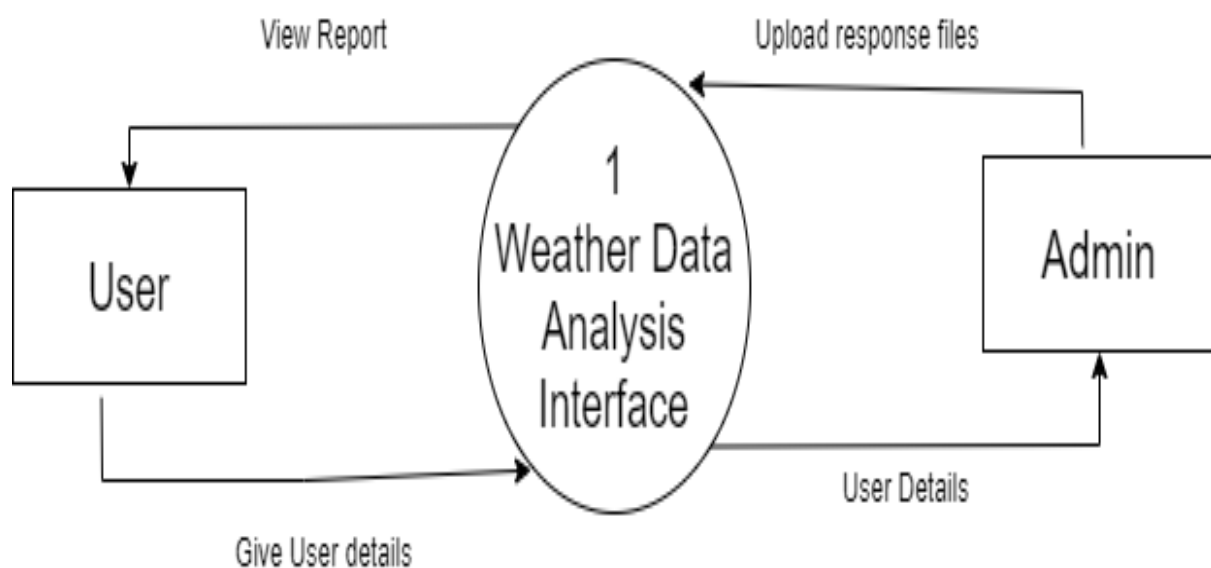
Level-1-DFD:
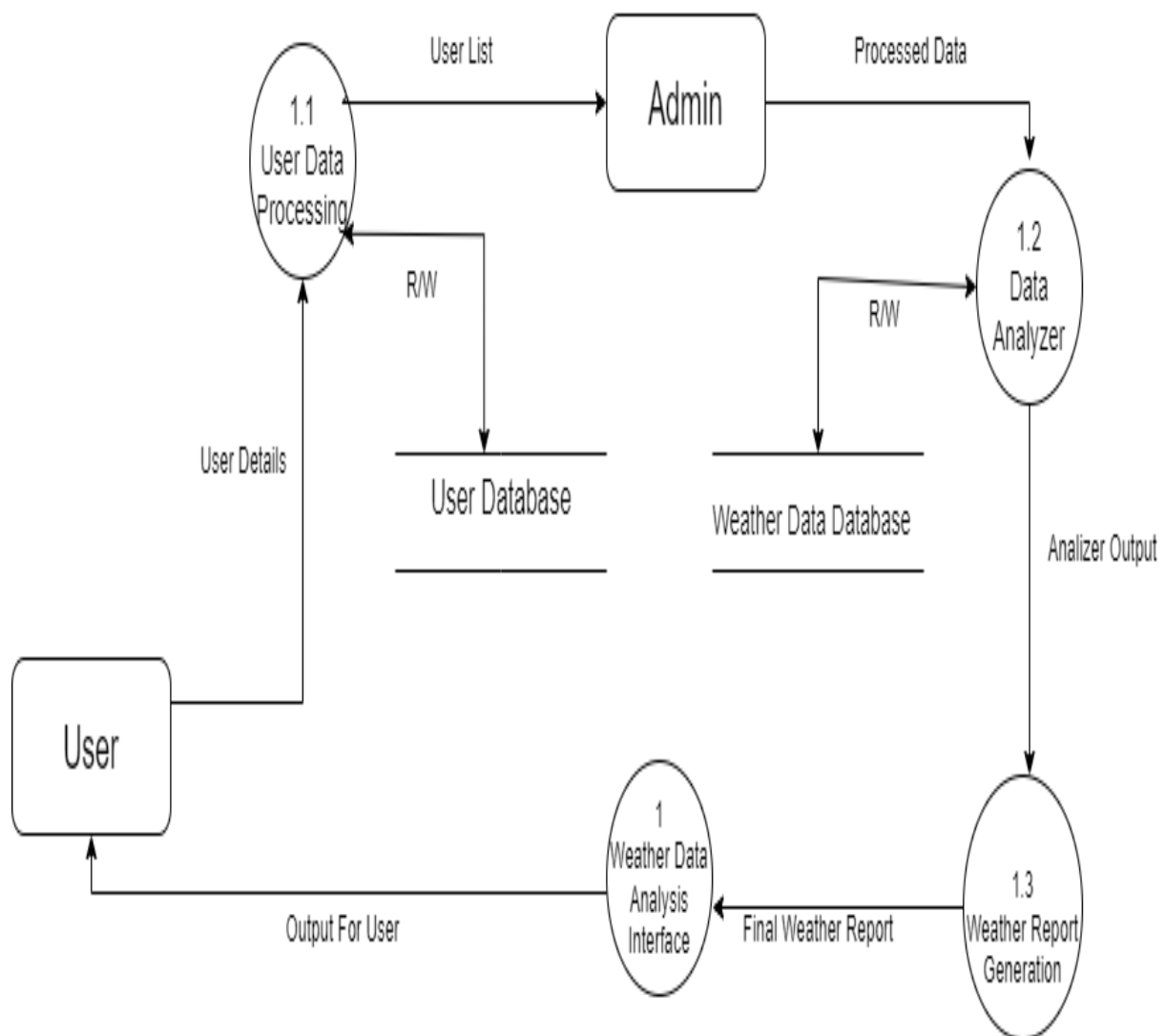
Level–2–DFD:
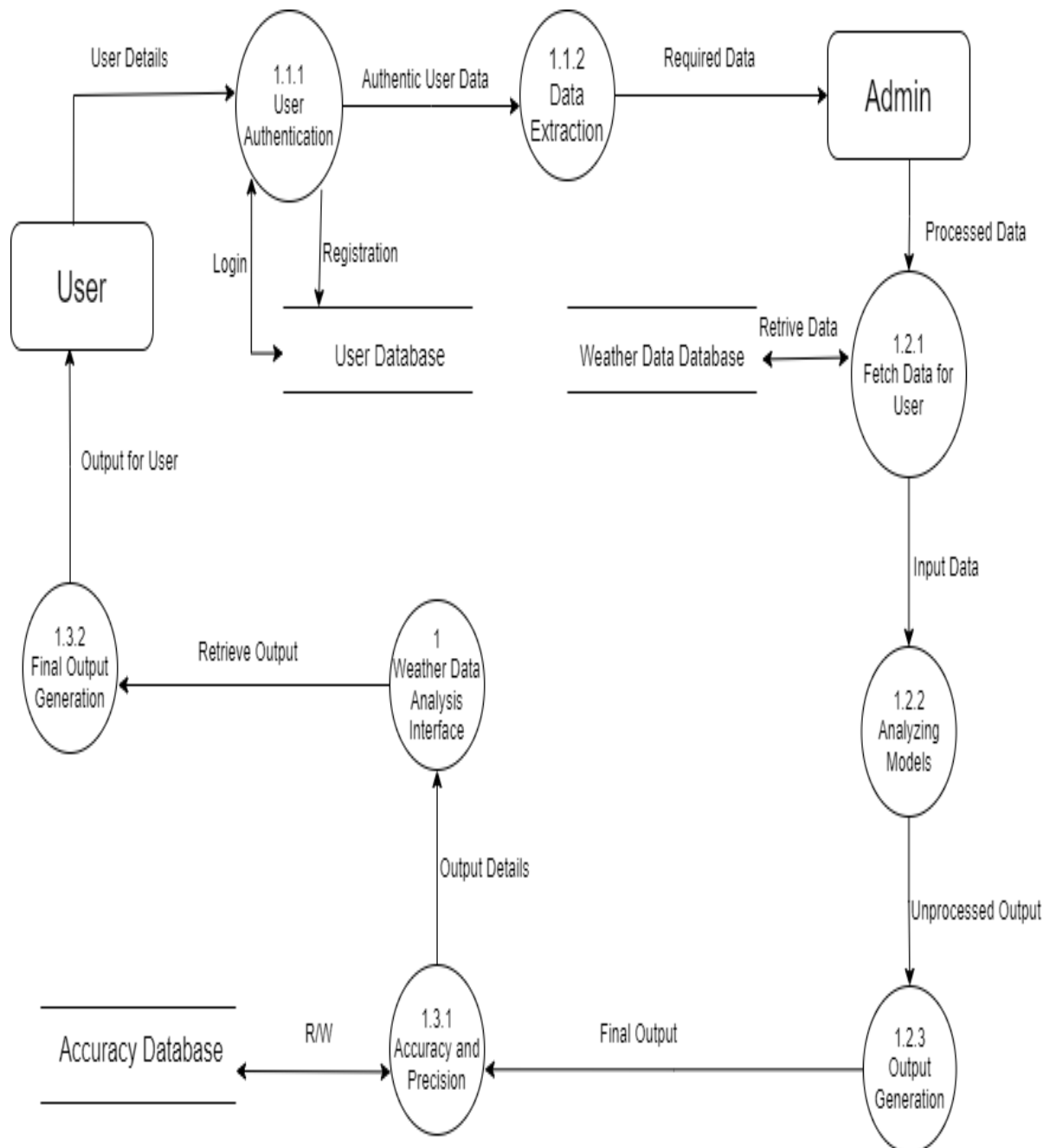
Conclusion:

The Software Requirements Specification (SRS) document for the Weather Data Analysis Application has outlined the essential requirements and specifications necessary for the successful development of this software solution. In this conclusion, we summarize the key points and emphasize the significance of the project.

The Weather Data Analysis Application is designed to serve meteorologists, researchers, weather enthusiasts, and decision-makers by providing them with a powerful tool for accessing, analysing, and visualizing weather-related information. It addresses the growing demand for accurate and timely weather data, which is critical for various industries and applications, including agriculture, transportation, disaster management, and climate research.

Throughout this SRS document, we have defined the scope, objectives, functional and non-functional requirements, external interfaces, and constraints of the application. The project aims to create a user-friendly, secure, and scalable system that leverages external data sources and APIs to provide real-time and historical weather data analysis.

The next steps in the project involve system architecture design, development, testing, and deployment. Close collaboration between the development team, stakeholders, and quality assurance experts will ensure that the application meets the specified requirements and exceeds user expectations.

This SRS document serves as a foundational blueprint for the Weather Data Analysis Application, guiding its development and ensuring alignment with the project's goals. As the project progresses, this document will be a valuable reference for stakeholders, project managers, and developers to ensure the successful delivery of a robust and valuable weather data analysis tool.