# AI-Driven Dengue Outbreak Prediction and Early Warning System


ABDULLAH RAZA
**01-134221-008**
FATIMA ZAKI
**01-134221-021**

Supervisor: Dr. Faisal Imran


**Bachelor of Science in Computer Science**


Department of Computer Science
Bahria University, Islamabad


October 2025

# Certificate

We accept the work contained in the report titled "Dengue Outbreak Prediction and Warning System", written by Mr. Abdullah Raza AND Ms. Fatima Zaki as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor:

______________________________

Internal Examiner:

______________________________

External Examiner:

______________________________

Project Coordinator:

______________________________

Head of the Department:

______________________________

# Abstract

Dengue fever continues to be a significant public health issue in Pakistan, mostly seen in cities like Lahore and Karachi, which face yearly outbreaks that put pressure on healthcare systems. Being able to predict these outbreaks in advance can facilitate timely interventions, minimize disease spread, and enhance resource allocation for public health officials. This project seeks to create a data-driven predictive model for forecasting dengue outbreaks by utilizing public health data, weather trends, population density, and geospatial analysis. which incorporates machine learning methods such as time-series forecasting and by utilizing datasets from opensource portals and climate data sources, the system will pinpoint high risk areas and provide warnings. Furthermore, a geospatial analysis module will be included to visualize regions and their probable dengue cases using GIS tools and heatmaps. A trained chatbot will also be developed to propose preventive measures. To enhance public health awareness and timely interventions an automated alert notification system will inform users when they enter certain high-risk areas about potential outbreak threats. The system will be accessible by an interactive dashboard and a mobile application to provide easy accessibility and a hands on user experience which will enable users to track dengue trends and implement proactive measures.

In addition to dengue case predicitons, the system emphasizes awareness through an intelligent, multi-platform interface. The integration of real-time risk mapping, Firebase powered push notifications, and a chatbot based on precautions which ensures that users receive personalized and timely information to make informed health decisions. By combining AI driven forecasting, GIS, and communication technologies, this project aims not only to support proactive dengue management but also to contribute toward the development of a scalable framework for digital disease surveillance in Pakistan.

# Acknowledgments

All praise is to Almighty Allah who bestowed upon us a minute portion of His infinite knowledge, by virtue of which we were able to complete this meaningful and challenging task.

We are deeply thankful to our supervisor Dr. Faisal Imran for his guidance, support, and insightful feedback throughout the development of our project. Without his continuous supervision and encouragement, the successful completion of this work would not have been possible.

We also extend our heartfelt gratitude to our parents and families, whose constant support, prayers, and belief in us have been a source of motivation and strength during this academic journey.

ABDULLAH RAZA
FATIMA ZAKI
Bahria University
E-8, Islamabad, Pakistan

October 2025

# Contents

# List of Figures

# List of Tables

# Acronyms and Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| DBMS | Database Management System |
| GIS | Geographic Information System |
| JSON | JavaScript Object Notation |
| REST | Representational State Transfer |
| FCM | Firebase Cloud Messaging |
| HTTP | Hypertext Transfer Protocol |

# Chapter 1

# Introduction

Dengue fever is a viral infection transmitted by mosquitoes which is a severe health concern to the population, especially in tropical and subtropical regions [1]. Pakistan has seen a high level of frequency in outbreaks over the years, which lead to serious health, social, and economic problems. Conventional mechanisms of disease surveillance are usually associated with delays in data reporting and inadequacy in real time predictive information, which limits the capability of the authorities to execute effective preventive measures.

Considering these difficulties, this project has a smart and information-based answer to determine possible areas of dengue outbreak and notify users accordingly. The system combines dengue risk prediction machine learning models with geospatial visualization, and enable users to single out areas that are considered high-risk on an interactive map. Furthermore, it includes chat bot services, push notifications, and dengue related precautionairy measures, delivering insights and both predictive and actionable health data to users.

The necessity of such a system is explained by the increasing influence of the outbreaks of dengue on civic health facilities and absence of proactive systems to avoid spread of dengue cases. An early warning system can facilitate the timely response, effective allocation of resources, better community awareness, and this would reduce the spread of disease and prevent contributions to large-scale ones By transforming raw data into actionable insights. This system supports authorities, healthcare workers, and the public in making informed decisions that can save lives.

In general, this project will enhance dengue prevention and control in Pakistan by applying artificial intelligence and the use of modern technology. It is user friendly and based on prediction, visualization and awareness which leads to the establishment of a better-prepared and more knowledgeable society able to react successfully to the challenges of future dengue.

## 1.1 Background

The spread of Dengue has been a growing issue in Pakistan because of lack of urbanization, sanitation and climate changes which give advantage to breeding of mosquitoes. Over Thousands of cases of dengue have been reported per year in the past decade, and there has been great significance in cases taking place in large cities like Lahore, Karachi, and Rawalpindi [2]. These outbreaks that keep reoccurring demonstrate the necessity of active surveillance and early warning mechanisms.

Health departments mostly rely on manual data collection methods. Their response is based on containment once an outbreak has already commenced. However, as digital sources of data such as weather data, population continue to become more accessible. dengue disease cases can be predicted by methods of AI modeling and can be used to predict the level of risk of dengue in advance of an outbreak.

Recent development in machine learning[3], geographic information systems (GIS), and web-based dashboards have made it possible to create intelligent platforms that are capable of analyzing diverse datasets and can provide visual representation of outbreaks. These technologies can be integrated to create a project that aims to replace the traditional dengue surveillance with an automated and predictive one providing a user interactive solution enabling the authorities and citizens to do something prior to an outbreak escalation.

## 1.2 Problem Description

Dengue epidemics have become a chronic health issue in Pakistan, and thousands of people are affected by it. Despite the availability of dengue case reports, the present surveillance and response systems are more reactive instead of proactive, resulting in the following:

- Postponed measures that lead to increased spread of diseases.
- Health institutions being overwhelmed by dengue sudden increases.
- A lack of public knowledge, resultant in unproductive preventive steps[4].
- Inefficient allocation of resources, leaving hospitals unprepared.

Moreover, traditional practices do not factor historic cases in predictive models which makes it totally reliant on reactive means leaving no room for predicting cases and precautionairy measures.

The proposed project is aimed at solving these problems by developing an AI-based prediction system able to forecast correctly and aid in timely health responses.

## 1.3 Objective

The key objectives of this project are as follows:

- To setup a predictive system, that exploits machine learning algorithms to discover possible areas of dengue outbreaks in advance.
- To offer a convenient online platform to visualize the real-time dengue risk levels with the help of interactive maps and statistical dashboards.
- To improve awareness and involvement of the population providing notifications, instructions and chatbot based suggestions regarding dengue prevention
- To assist the management of decisions and planning in the public health departments through the provision real-time information and early alerts of potentially dangerous areas.

## 1.4 Scope

This project will be aimed at developing a detailed Dengue Outbreak Prediction and Warning System which relies on machine learning, geospatial analysis, and real-time alerts to issue early warnings of the high-risk regions in Pakistan. The system includes. The system includes a **web dashboard** to visualize data and a **mobile application** for user interaction and alerts. This project is aimed at helping the people and government to engage in preventive action against dengue outbreaks.

### 1.4.1 Included in the Project

- Compiling and examining past cases of dengue and weather data.
- Dengue cases prediction with a machine learning model(prophet).
- Conducting geospatial analysis to display the risk zones on an interactive map of Pakistan.
- Creating a Web dashboard (React.js) and a mobile app (React Native) for graphical representation, statistics, and chatbots.
- Firebase Cloud Messaging (FCM) for location based notifications.
- Enabling users or health administrators to download case data of districts, provinces or national level in CSV format.

### 1.4.2 Not Included in the Project

- Real-time mosquito density or field surveillance data due to limited data availability.
- Direct control or mitigation of dengue spread, the system only provides predictive insights and alerts.
- Integration with hospital or laboratory record systems.

# Chapter 2

# Literature Review

A literature review is a critical summary and analysis of existing research, studies, and published information relevant to a particular topic or field of study. It helps in identifying what is already known, what methods or technologies have been used, what gaps still exist, and how the current project or research fits into or builds upon previous work. In the context of this dengue prediction project, the literature review focuses on analyzing past research related to the spread and control of dengue fever, the use of machine learning in epidemiological predictions, and the role of geospatial mapping and historical data in early warning systems.

Dengue fever is one of the fastest-spreading mosquito-borne diseases, primarily affecting tropical and subtropical regions. Despite most cases being mild, severe manifestations such as dengue shock syndrome and dengue hemorrhagic fever can be fatal, with case-fatality rates as high as 20 percent in the absence of timely diagnosis and treatment [5]. The infection of more than 100 million people was estimated in 2017 alone, placing a burden on the population. Tremendous health, social and economic costs on the host nations[6].The increasing cases of dengue required urgent need for accurate outbreak prediction systems. predicting dengue cases and trends early can reduce further spread and potentially save lives.

## 2.1 Traditional Dengue Forecasting Methods

Previously, dengue prediction was mostly based on the conventional statistical models in order to simulate past incidence of dengue and seasonal patterns. Autoregressive Models such as Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA) and One of the most commonly used models Exponential Smoothing State Space Models (ETS) were used for predictions in early studies. These approaches proved helpful in detecting the periodicity of uncovering outbreaks and predict short-term number of cases, especially in the areas where surveillance data is consistent.

Gradually, scientists identified the need to use climate covariates such as temperature, humidity and rainfall, which have a direct impact on the breeding of the mosquitoes and virus transmission. This brought up more evolved forms of conventional models like Vector Autoregression (VAR) and Seasonal ARIMAX (SARIMAX).

The comparative analysis shows that traditional statistical models are still useful in short and interpretable forecasting, but their weakness is felt when compared to machine learning methods. The latter is able to record non-linear dynamics, assimilate more inputs and adjust to changing transmission patterns. However, knowledge of the traditional models was important in creating the initial groundwork. to predict epidemiologic processes, provide meaningful information on the time and seasonal trends before implementing machine learning[7].

## 2.2 Predictors used in Dengue Forecasting

The majority of the prediction models are highly dependent on climate features including temperature, rainfall and humidity .Nevertheless, it has been found that reliance on climatic data might not give a comprehensive picture of transmission.

- Some are based on climate variables only[8].
- Others may include vector characteristics [9].
- Some of them include demographic factors which are based on population [10].

Even though methods have improved, there is still no agreement on the universally best model.It is difficult because methods are mainly based on statistical models based on historical case reports to identify the high-risk areas precisely.

## 2.3 Modeling Approaches

Pediction models are primarily divided into two types:

1. **Statistical Models**
   - Linear regression
   - Poisson regression
   - Time series models
   - Generalized additive models

2. **Machine Learning Models**

    - Neural networks
    - Random forest
    - Ensemble methods

Recent studies indicate machine learning methods often outperform regular statistical models due to their ability to capture non linear relationships and capabilities to utilise larger datasets. [11].

## 2.4 Data Sources and Predictors

Most dengue models have drawn incidence data from national surveillance systems, along with hospital and laboratory reports. Climate variables remain the most widely used predictors, including temperature, humidity, rainfall, wind speed, and sunshine. Some studies have also explored environmental and climate change indices such as ENSO, SOI, and ONI [12]. A smaller subset has used vector indices (container index, Breteau index, mosquito infection rate) and demographic factors (population density, urbanization, water access, GDP per capita) [13].

Although diverse predictors have been tested, most models still rely mainly on climate data, with fewer integrating vector and socio-demographic variables. Importantly, no existing model combines all four categories comprehensively.

## 2.5 Existing Systems

Dengue fever has become one of the major challenges to the wellbeing of the people especially in crowded city centres such as Lahore, Karachi. This brings about the need of development of predictive models in order to forecast outbreaks and make timely interventions. Over the past years significant research has been done on dengue prediction using entomological indices [10], climatic variables [14], meteorological and socio-economic factors. In multiple studies Aedes aegypti and Aedes albopictus were reported to be significance in casuses of dengue fever [15]. Nevertheless, much researches have been done on dengue forecasting, yet there are still gaps that are critical on how to integrate geospatial analysis[16], research-driven recommendations, and real-time notification systems to enhance case predictions and precautionairy actions. Traditional methods mostly depend on statistical models using historical case reports, making it difficult to locate high risk areas precisely. Existing dengue alert systems are typically reactive, relying on news reports or hospital data when the cases have already taken place. These methods often fail to provide early warnings which leads to delayed public awareness and response. Moreover,

Moreover, most lack user interaction features (i.e. chatbots or awareness tools)[17] that may aid in educating. This puts out an obvious gap that our project seeks to address by creating dengue case prediction system with interactive, user friendly tools such as a web based dashboard and an AI-powered chatbot.

## 2.6 Challenges in Current Models

Generic dengue control methods, such as surveillance and community interventions, are constantly challenged by lacking urbanization, climate variability, and insufficient infrastructure in rural areas. The transmission of dengue is spatially heterogeneous because of variations in vector habitats, changes in climatic conditions, and human activities. The interaction between human, climate, and mosquito factors determines how outbreaks occur.[10]. There are numerous existing models, which differ in purpose and settings, but some of them are based only on climate. Other variables, include the use of vectors or population properties. However, inaccuracies are because of geo-spatial differences, inadequate surveillance information, and inability to count because of time lags in disease dynamics, which may delay predictions and overwhelm health systems.

## 2.7 Emerging Trends

In recent years, dengue prediction research has increasingly prioritised the use of machine learning (ML). Techniques such as random forests, neural networks, boosting methods have shown improved prediction accuracy compared to generic statistical models. These methods are effective at handling large datasets that capture non linear relationships between variables which are often ignored by generic statistical approaches.

## 2.8 Research Gaps and Motivation

The gaps identified in the reviewed literature are the following ones:

- Lack of unified models, which integrate climate, demographic and environmental features.
- The current dengue warning mechanisms are generally reactive.
- Lack in reporting of model performance and inadequate external validation.

These weaknesses demonstrate the necessity of enhanced, evidence-based, and validated dengue prediction system. The following challenges are considered in our project through integration of dengue prediction using historic case data and various climate factors along with demographic fators.

Table 2.1: Comparison of Related Dengue Forecasting Studies

| **Study / Author (Year)** | **Model Type / Method** | **Predictors Used** | **Key Findings / Limitations** |
|---|---|---|---|
| Hii et al. (2012) [18] | Poisson regression / time-series | Climate | Forecasts 16 weeks ahead and misses non-weather factors. |
| Eastin et al. (2014) [19] | Autoregressive time-series | Climate, Historical data | Good at using past patterns; may miss quick changes or non-straight trends. |
| Nguyen et al. (2022) [20] | Deep neural network models | Climate | Better accuracy in predictions; needs lots of data and watch for errors. |
| Adde et al. (2016) [12] | Regression model | Climate | Predicts 80% of outbreaks; doesn't include virus types or immunity. |
| Johansson et al. (2016)[11] | Time-series model | Climate | Good for case prediction; struggles with sudden changes. |

# Chapter 3

# Requirement Specifications

System requirements are configurations that are necessary for proper functionality. Not fulfilling these requirements may lead to poor performance. This chapter states the functional and non functional requirements, along with the software specifications necessary for successful development. Moreover, the chapter also includes use case diagrams to show the interactions between users and the system.

## 3.1 Proposed System

In the specified system, the first step involves collecting relevant environmental and health related data such as rainfall, humidity and historical dengue case records. This data is then used to train a machine learning model to identify trends and patterns to predict dengue outbreaks in different geographical locations across Pakistan. Once predictions are made the system highlights areas on an interactive geospatial map based on risks assigned to them and is then embedded within a userfriendly dashboard on web along with a mobile application. Furthermore, the system contains a chatbot feature to assist users. The end goal is to enable timely awareness and assist health departments in proactive resource allocation and mosquito control efforts.

## 3.2 Intended Audience

Target users include:

- **Public Health Officials** – for monitoring and preparing precautionairy measures.
- **General Public** – to be aware or potential dengue risks.

- **Researchers and Analysts** – to analyse outbreak patterns and utilise the information in improvement of further studies.

## 3.3 User Classes and Characteristics

### 3.3.1 Identify Requirements

This project uses machine learning model named Prophet to predict dengue cases outbreaks. software components include Python and Node.js for backend development and for APIs. Frontend built using React.js and react native. GIS libraries (like Leaflet) used for geospatial representations.

### 3.3.2 Feasibility Report

The feasibility of the Dengue Prediction System is increased by the digital access and the availability of open source climate and health related data. Technical feasibility is achieved by the use of reliable libraries, while financial feasibility is achieved by using free APIs where possible.

## 3.4 Software Requirements

### 3.4.1 Client Side

- **Web App:** React.js, Leaflet.js, modern browser (Chrome/Edge/Firefox)
- **Mobile App:** React Native (Expo), Firebase SDK (notifications), Android Emulator / iOS Simulator

### 3.4.2 Server/Backend

- **OS:** Windows/Linux
- **Backend:** Node.js (v18+), Express.js
- **Database:** MongoDB

### 3.4.3 Machine Learning

- **Libraries:** Python 3.10+, Prophet, Pandas, NumPy, Scikit-learn
- **Model:** Prophet to forecast dengue trends and get predictions through API.
- **Chatbot:** Fine-tuned Mistral model on Colab and deployed on Hugging Face, accessed via API.

## 3.5 Functional Requirements

Functional requirements describe the functional aspect of the system and how user will use the functionality of the system. Following are the functional requirements of the system with respect to user.

### 3.5.1 Search District Risk

- Allows users to search and view dengue risk predictions for any district in Pakistan.
- achieved by fetching predictions from the backend through APIs.
- Display the predicted risk level (High, Medium, Low) along with predicted cases.
- Ensures clear visualisation.

### 3.5.2 View Risk Map

- Displays a geospatial map of Pakistan using Leaflet.js using GeoJSON data of the districts.
- Highlight each district based on dengue risk levels (red for high, orange for medium, green for low).
- Allows users to click on any district to view information in a pop up such as predicted case count and risk level.

### 3.5.3 Chatbot Support

- Provides a chatbot trained with dengue related dataset through Mistral model on Hugging Face.
- Allow users to ask questions related to dengue.
- Offers a user friendly responses so that the user may easily understand the response.

### 3.5.4 Push Notifications

- Send daily dengue alert notifications to users when their district is at high risk by using Firebase Cloud Messaging (FCM).
- Alerts includes a warning to notify about the high risk.
- Ensures public safety through backend triggered notification scheduling.

### 3.5.5 View Statistics

- Displays dengue trends using graphs and charts of district, provincial, and national levels.
- Allows filtering results by date to observe outbreak patterns.
- Enables data export functionality, allowing users to download prediction results in CSV format for further research or documentation.
- Supports visual analytics for better understanding of outbreak progression and prediction model performance.

### 3.5.6 Guidance Information

- Provides separate informational tabs for dengue symptoms and preventive measures.

## 3.6 Non-Functional Requirements

Non-functional requirements of the system describe the capabilities of the system and how it should work. Following table shows the non-functional requirements of the proposed system:

Table 3.1: Non-Functional Requirements of the System

| Sr. No | Non-Functional Requirement | Category |
|---|---|---|
| 1 | The platform should deliver prediction results and map loading within **5 seconds** | **Performance** |
| 2 | The user interface must be intuitive for both web and app and easy to navigate for both health officials and the general public. | **Usability** |
| 3 | The system must be capable of handling **concurrent users** without performance degradation. | **Scalability** |
| 4 | Secure APIs and Protect user location and personal data | **Security** |
| 5 | System should be available 24/7 with minimum downtime. | **Availability** |
| 6 | The ML model must be easily replaceable or upgradable without disrupting the entire system. | **Modularity** |

## 3.7 Use Case Diagram

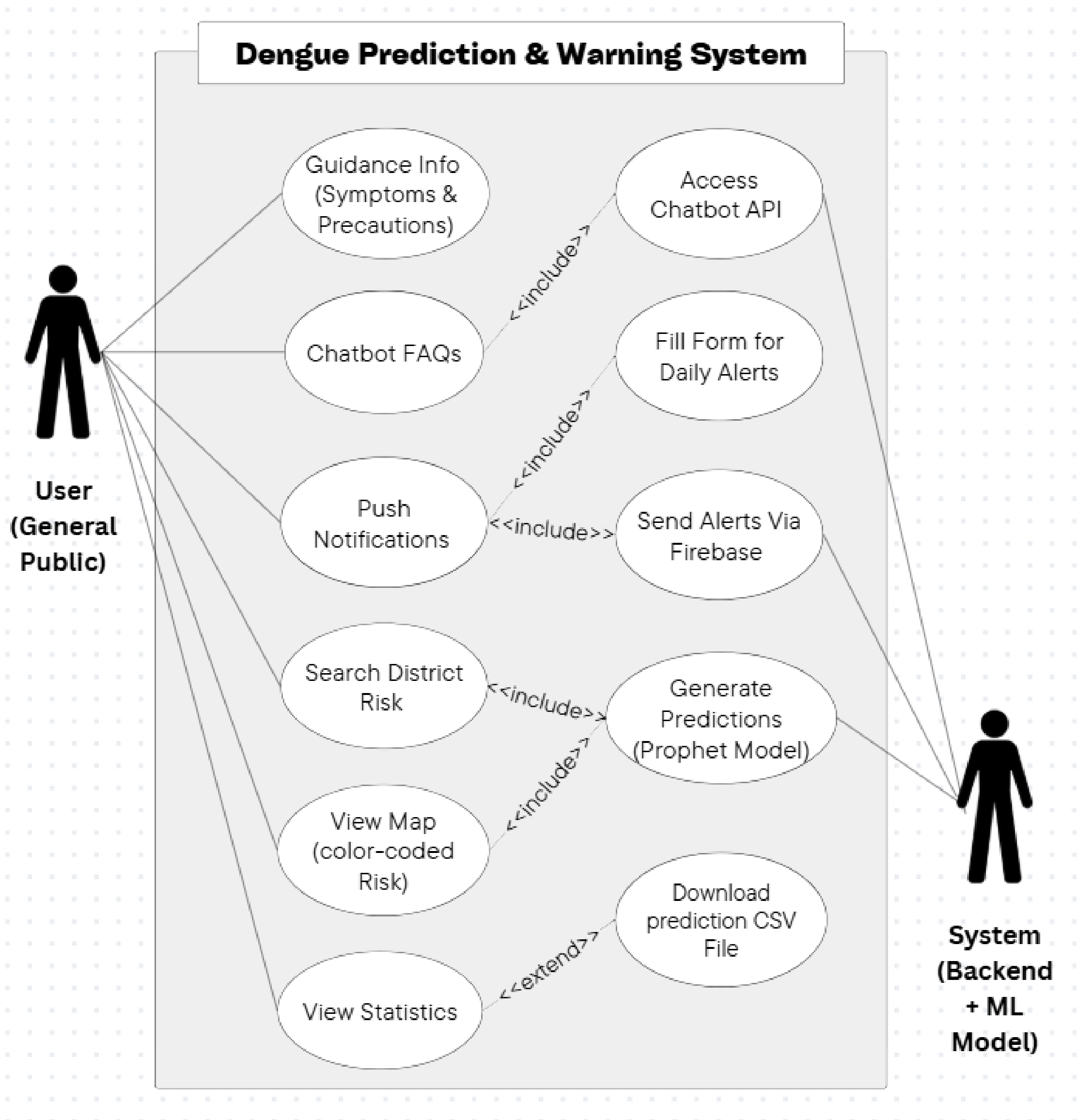


Figure 3.1: Complete Use Case Diagram

The use case diagram for this system models the interactions between the end user and the core functionalities of the Dengue Prediction and Alert System. Since there is no admin module, the primary actor is the public user, who can access prediction results, receive alerts based on their location, interact with the dengue guidance chatbot, and view risk zones on a map, secondary actor is system who performs actions independently of the user (like generating risk data, and triggering alerts).

## 3.8 Use Cases

### 3.8.1 Search District Risk

This use case allows users to search for dengue risk predictions for any district across Pakistan. When a user enters the district name in the search bar, the system retrieves relevant data from the backend API, which is connected to the trained machine learning model. The result includes risk levels such as high, medium, or low with estimated case counts. The feature enables users to quickly assess the risk in their selected region allowing timely awareness.


Figure 3.2: Search District Risk

Table 3.2: Use Case: Search District Risk

| | |
|---|---|
| **Use Case ID:** | 1001 |
| **Use Case Name:** | Search District Risk |
| **Actor:** | User |
| **Description:** | User searches for a district to view its dengue risk level and view its predictions. |
| **Trigger:** | User enters a district name in the search bar. |
| **Preconditions:** | Ai predictions must contain district risk data. |
| **Post Conditions:** | User sees predicted risk and case details for the selected district. |
| **Normal Workflow:** | 1. User enters district in search bar.<br>2. System calls API to get predictions for matching district.<br>3. System retrieves and displays risk information. |
| **Alternative Workflow:** | 1. If prediction unavailable or API doesn't work, system shows "Risk Unknown." |

### 3.8.2 View Risk Map

The Risk Map allows users and health officials to visually explore dengue predictions through a highlighted map of Pakistan. The map highlights each district highlighted colors representing different risk levels for example red for high, yellow for medium, and green for low risk. Users can click on any district to view a popup showing predicted case numbers and risk level. This feature helps in identifying outbreak locations, enhancing awareness and decision making for public health planning.

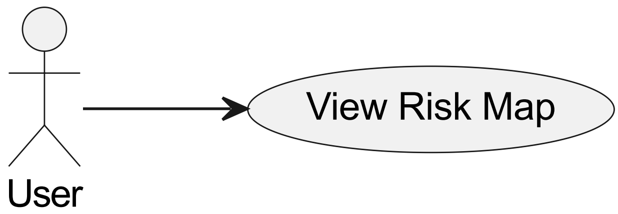


Figure 3.3: View Risk Map

Table 3.3: Use Case: View Risk Map

| | |
|---|---|
| **Use Case ID:** | 1002 |
| **Use Case Name:** | View Risk Map |
| **Actor:** | User |
| **Description:** | User views color-coded dengue risk map. |
| **Trigger:** | User opens the map tab or dashboard. |
| **Preconditions:** | Map and district-level risk data must be loaded. |
| **Post Conditions:** | User sees an interactive map with highlighted high/medium/low risk zones. |
| **Normal Workflow:** | 1. User opens the map dashboard.<br>2. System loads the Pakistan's map.<br>3. System shows the risk using colored highlights over the districts.<br>4. User clicks a district of choice.<br>5. System shows a popup with predicted cases and risk level. |
| **Alternative Workflow:** | 1. If map data not available, system shows error message. |

### 3.8.3 Receive Alerts (Push Notifications)

This use case enables users to receive dengue risk alerts through Firebase Cloud Messaging (FCM). When users subscribe by entering their name and city, the system saves their information and checks for dengue prediction data for that location periodically. If the model detects a high risk condition in the subscribed area, the backend triggers a notification alert that appears on the user's mobile or web browser. This displays a pop up notification with warnings and helps users take timely precautions.

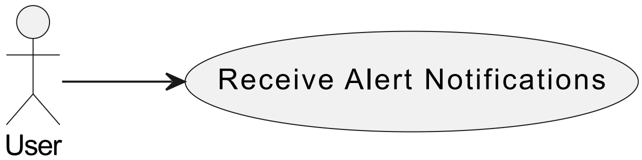


Figure 3.4: Receive Alerts (Push Notifications)

Table 3.4: Use Case: Receive Alerts (Push Notifications)

| | |
|---|---|
| **Use Case ID:** | 1004 |
| **Use Case Name:** | Receive Alerts (Push Notifications) |
| **Actor:** | User |
| **Description:** | User subscribes to receive push notifications for dengue alerts in their city. System sends notifications via Firebase Cloud Messaging. |
| **Trigger:** | User submits subscription form (name, city) OR system detects high-risk in city. |
| **Preconditions:** | User must submit subscription form. Firebase Cloud Messaging must be configured. |
| **Post Conditions:** | User receives push notification when their city is at risk. |
| **Normal Workflow:** | 1. user opens the subscription form.<br>2. User submits the form.<br>3. System saves user name and city in database.<br>4. System subscribes user to notifications for that city.<br>5. When city enters high risk the system triggers Firebase Cloud Messaging.<br>6. User receives a push notification. |
| **Alternative Workflow:** | 1. If form incomplete, system shows error message.<br>2. If FCM fails notification will not be received.<br>2. If app is foreground show in app banner notification.<br>2. If app is closed push notification is sent. |

### 3.8.4 Chatbot Support

The Chatbot provides users with an interface to ask dengue related questions. Built using the Mistral model deployed on Hugging Face, the chatbot offers AI driven responses to queries related to dengue symptoms and prevention. This is useful for general users who seek quick and reliable information.

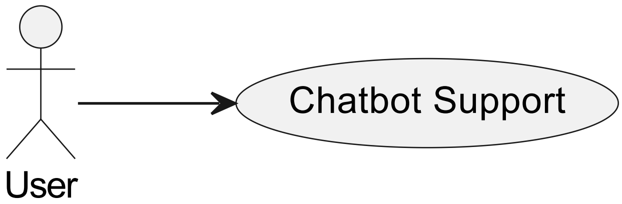


Figure 3.5: Chatbot Support

Table 3.5: Use Case: Chatbot Support

| | |
|---|---|
| **Use Case ID:** | 1005 |
| **Use Case Name:** | Chatbot Support |
| **Actor:** | User |
| **Description:** | User interacts with dengue chatbot for FAQs. |
| **Trigger:** | User starts chatbot session. |
| **Preconditions:** | Chatbot model and API must be active. |
| **Post Conditions:** | User receives appropriate response. |
| **Normal Workflow:** | 1. User opens chatbot.<br>2. System loads chatbot interface.<br>3. User asks dengue-related question.<br>4. Chatbot returns response. |
| **Alternative Workflow:** | 1. If chatbot unavailable, system shows “Chatbot Error.” |

### 3.8.5 View Statistics

This use case allows users and health authorities to analyze dengue outbreak patterns through graphs and charts.The data is displayed at national, provincial, and district levels. Users can filter data by specific date ranges to view seasonal trends. Furthermore, the feature enables users to download the displayed prediction data in CSV format for further analysis.

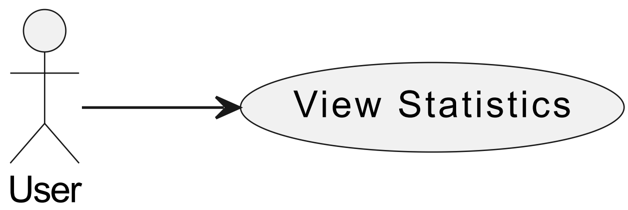


Figure 3.6: View Statistics

Table 3.6: Use Case: View Statistics

| | |
|---|---|
| **Use Case ID:** | 1006 |
| **Use Case Name:** | View Statistics |
| **Actor:** | User / Public Health Official |
| **Description:** | Analyze dengue trends through graphs and charts at district, provincial, and national levels. Users can filter results by date and download the displayed prediction data as a CSV file. |
| **Trigger:** | User opens the statistics tab. |
| **Preconditions:** | System must have stored dengue history and access to prediction API. |
| **Post Conditions:** | User sees interactive visual graphs and charts. |
| **Normal Workflow:** | 1. User opens the statistics tab.<br>2. System loads graphs and charts (province/district/national level).<br>3. User selects the date range.<br>4. System updates and show graphs.<br>5. User can download predictions. |
| **Alternative Workflow:** | 1. If data not available, system shows “Statistics not found.” |

### 3.8.6 Read Awareness and Guidance Info

The Read Awareness and Guidance Info use case offers dengue educational materials. Using specific tabs with the titles of Symptoms and Precautions. This section is used to guide users to recognize early warning signs of dengue. Precautions to prevent the chances of infection.

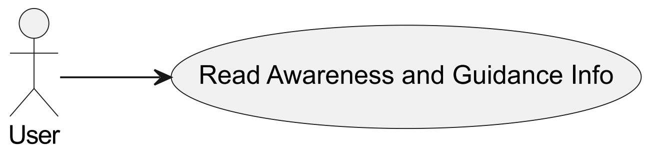


Figure 3.7: Read Awareness and Guidance Info

Table 3.7: Use Case: Read Awareness and Guidance Info

| | |
|---|---|
| **Use Case ID:** | 1008 |
| **Use Case Name:** | Read Awareness and Guidance Info |
| **Actor:** | User (general public) |
| **Description:** | User reads dengue-related awareness material such as symptoms and precautions via dedicated tabs. |
| **Trigger:** | User clicks the “Symptoms” or “Precautions” tab. |
| **Preconditions:** | System must have awareness content stored. |
| **Post Conditions:** | User views the awareness and guidance information. |
| **Normal Workflow:** | 1. User selects “Symptoms” or “Precautions” tab.<br>2. System displays the content on screen. |
| **Alternative Workflow:** | 1. If content unavailable, system shows “Content not available.” |

# Chapter 4

# Design

System design contains architecture, components, and modules that are required to create a functional and reliable dengue prediction and alert solution. Chapter presents the design of Dengue Outbreak Prediction and Warning System. It provides a description of the general system flow, the architecture and development model used and the detailed components of design that combine to create the entire system. It aims to tell how different modules such as prediction model (Prophet), chatbot (Mistral), Firebase notification service and interactive dashboards work together to provide predictions of dengue accurately, user awareness and timely notifications via mobile applications.

## 4.1 System Architecture

The system architecture of the Dengue Outbreak Prediction and Warning System, a solid understanding of the interactions between various components of the system how they interact and perform. The architecture is further subdivided into various layers at the topmost level, which are data input, prediction and processing, application and backend services, data visualization and reporting, and user interaction and alerting. The functions carried out by each layer are different and contribute to the overall performance of guaranteeing real time monitoring, proper prediction and effective communication of users on the dengue risk levels.

the data input layer, there are data and weather collection of verified health sources. These datasets are the basis of making predictions. The prediction and processing level will involve the Prophet machine learning model which will be used to determine the trends of future dengue outbreaks based on the data collected. The Prophet model is used to predict the cases of dengue in the provinces on a monthly basis. A weighted disaggregation technique is used in estimating the outbreaks in the district level by taking into consideration the socio-environmental factors, including population density, rainfall, and HDI. The relative risk of each district is estimated with reference to these normalized indicators and provincial forecasts are allocated between districts proportionally in each province based on their risk scores. The trained model is availed via an API in order to make it scalable and efficient.

The back end services layer is established on the basis of the Node.js that serves as the main communication nodes between the prediction model, database, and the front end interfaces. It handles APIs, executes logic to fetch and send data and background services like push notifications via Firebase, which is scheduled via cron jobs. The web app and mobile application share the same backend which is accessed via APIs to ensure consistency and efficiency. The web application and the mobile app are built based on React.js and React Native (Expo) respectively as the data visualization and reporting layer. It offers users with a more interactive and intuitive interface in the form of color-coded maps, risk analysis charts and statistical graphs. Users can also see national, provincial, or district-level data, choose their own time ranges, and they can also download prediction reports to be further analyzed or documented

. Lastly, user interaction layer allows users to interact with the system in a number of ways. Mistral model is the chatbot that can answer dengue-related questions like symptoms, precaution, and treatment recommendations and is operated by Hugging Face. Location-based notifications are also provided in the system where they send notifications to the user if they are in a high-risk zone. These notifications are sent daily so that the user can do measures to stay safe. This multilayer system is designed to assure that such an architecture will not only anticipate the occurrence of dengue epidemics but will also enable people to be aware and take action. Its scalable design can be extended through the modular design in the future with additional secondary features like hospitals data connectivity, advanced visualization, and AI-based health recommendations.

### 4.1.1 High Level System Architecture Diagram

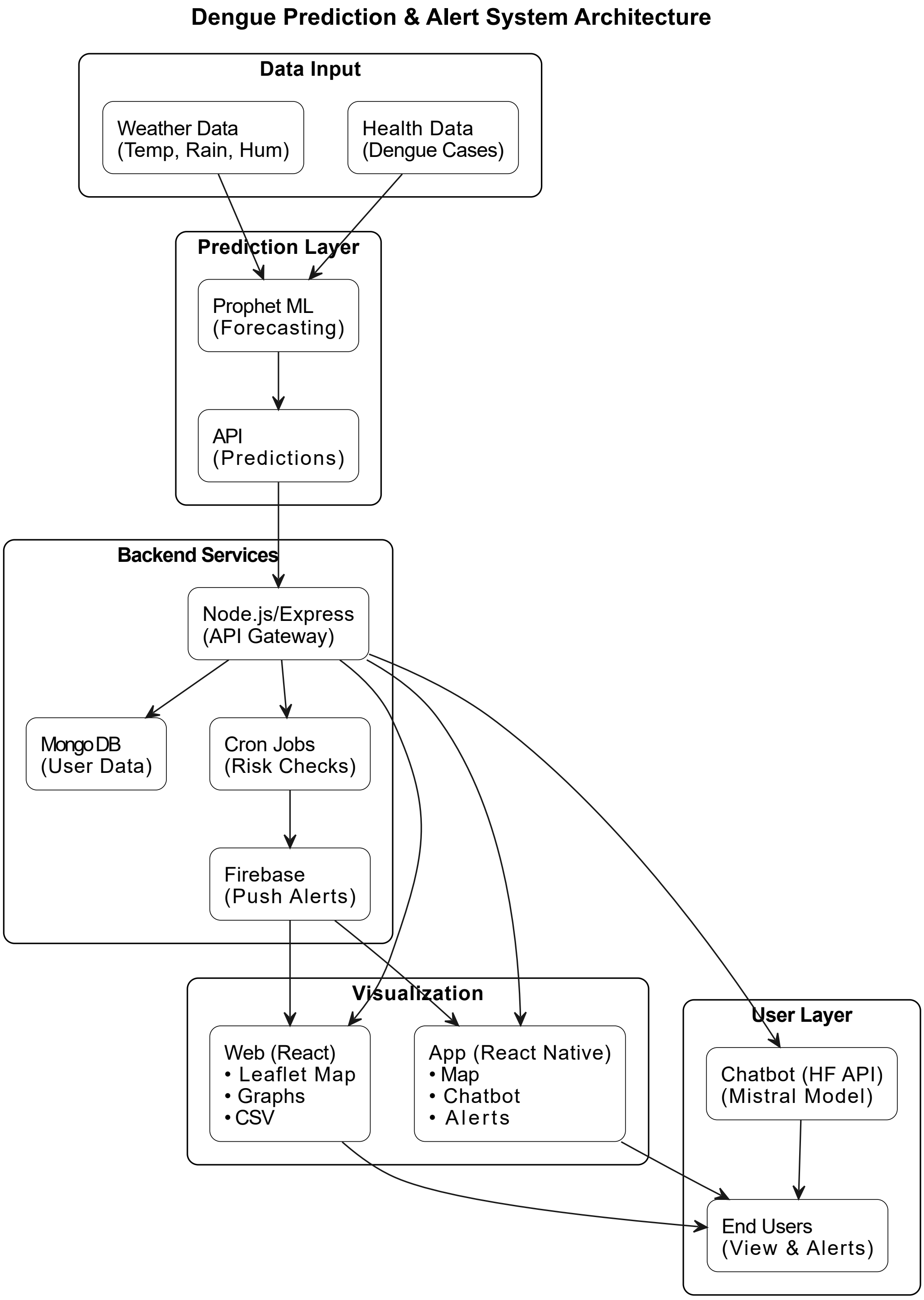


Figure 4.1: High Level System Architecture Diagram

It is a high level system architecture diagram, which shows interconnected layers that provide data collection, prediction, visualization, and alert. These elements collectively offer predictions and alerts to users in a timely manner.

### 4.1.2 Software Architecture

The software architecture diagram gives a general picture of the whole DenguePredict system showing how its main components interact with each other to produce predictions, warnings, and help to the users. It has a top to bottom layered design beginning with the primary application and cascades downward through the user interfaces, backend service, prediction pipeline, data storage, and external integrations. The logical packages are clearly divided into each component like the Node.js API, Prediction model, MongoDB, Firebase Cloud Messaging and chatbot API to demonstrate its place in the system. The flow of data among modules is also noted by the diagram and indicates that weather data, historical records of dengue, and GeoJSON files are used to facilitate the forecast. These interconnections help visualize the structure of the system by providing a blueprint to the system, which is to be understood in order to be clear to the developers, stakeholders, and evaluators..

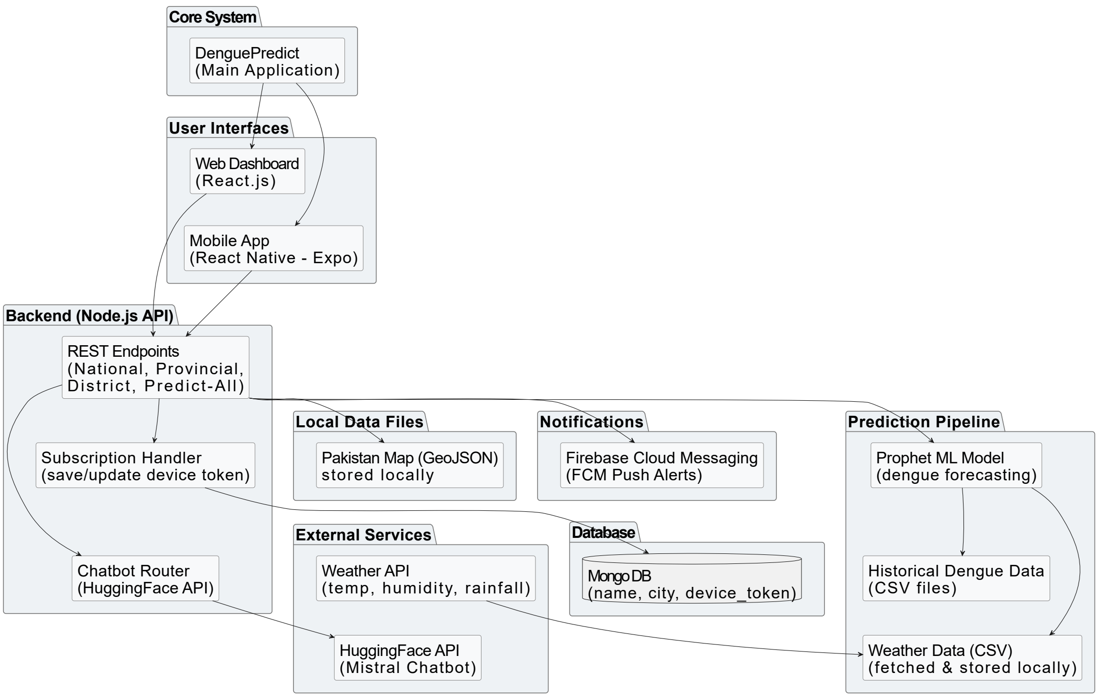


Figure 4.2: Software Architecture Diagram

### 4.1.3 Model Process Diagram

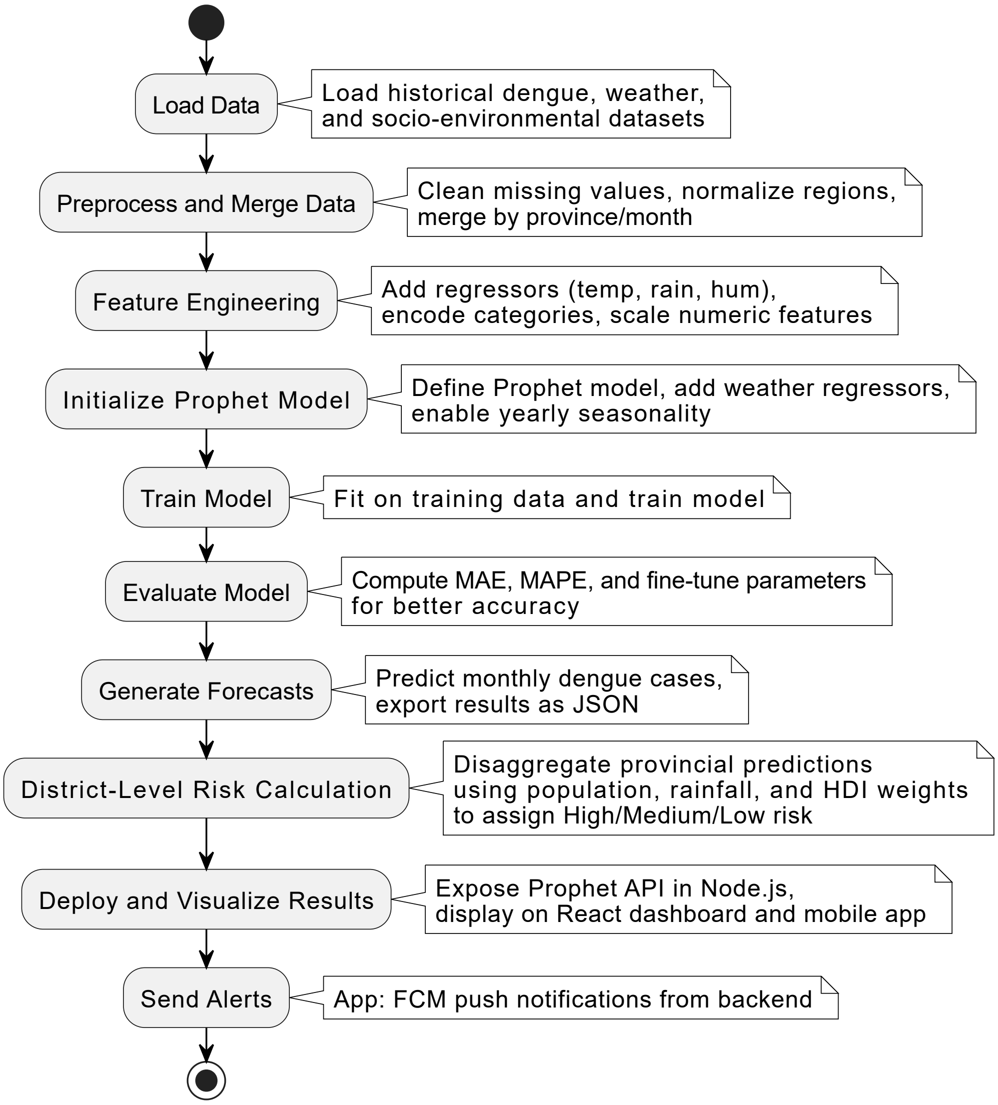


Figure 4.3: Model Process Diagram

## 4.2 Design Methodology

DenguePredict system was developed in an Agile, iterative manner, which provided flexibility, as different modules were developed with time. The first step of the project was the basic prediction pipeline based on the Prophet model and the pre-processing of historical data on dengue and weather. Then, the Node.js server was built to make both the web dashboard and mobile app have structured REST APIs and can help in seamless interaction and data flow.

Later versions saw the creation of React web dashboard and React Native mobile application with Firebase Cloud Messaging being added to enable push notifications Other modules like the chatbot, geospatial visualization based on GeoJSON, and automated cron-based notifications were also introduced in other cycles.

The iterations involved development, testing, integration, and refinement parts, so the accuracy of prediction, system reliability, and user experience were continuously improved.

## 4.3 Design Constraints

The constraints of the Dengue Outbreak Prediction and Early Warning System design dictate the design to be feasible, maintainable, and accessible to the users in both web and mobile platforms. These boundaries are what steer the development process and contribute towards system boundaries and constraints.

### 4.3.1 Exclusions

- **Offline Functionality:** The system must be connected to the internet to receive prediction information, rainfall information, and API-based insights.
- **Manual Data Entry:**Prediction or case data can not be changed manually by the user, all data is retrieved via preprocessed datasets and APIs.
- **Medical Diagnosis:** The system lacks any medical advice other than precautionary measures or suggestions in regards to dengue related queries only.
- **User Authentication:** There will be no login and registration features to keep the process simple.

### 4.3.2 Assumptions

- Users will subscribe to get area-based and accurate dengue alerts.
- Accuracy of prediction depends on the quality of past case information and prediction of rains.
- The users possess fundamental digital literacy to operate web dashboards or mobile dashboards.

## 4.4 High Level Design

High-Level Design gives a general idea of the interactions between the key modules and functionality of the Dengue Outbreak Prediction and Warning System

### 4.4.1 Presentation Tier (Frontend)

A web dashboard (React.js) and a mobile app (React Native + Expo) are part of the frontend. The web dashboard is presented in the form of an interactive color-coded map, date filters and the possibility of a CSV download, chatbot access, and awareness tabs. The mobile application will contain a home screen where a user can use Map, Search, Statistics, Chatbot, Subscribe, Symptoms, and Precautions and a subscription form based on Firebase Cloud Messaging (FCM) to get push notifications.

### 4.4.2 Application Tier (Backend)

Applied in Node.js/Express, the back-end provides REST APIs to predictions, statistics, chatbot, and subscription. MongoDB is used to store subscription data and only one active subscription is maintained per device token. The node-cron jobs track the risk level of the district and send FCM alerts when a user is at high-risk in his or her city.

### 4.4.3 Prediction & Analytics Tier

A Prophet model with the historical data of dengue and weather regressors is used to perform a time-series forecasting. Using the socio-environmental factors, provincial forecasts are disaggregated to the district level. Prediction APIs are served by reading forecasts (local JSON files: provincial-dengue- forecasts.json, district-dengue-forecasts.json).

### 4.4.4 Data Tier

Data tier contains MongoDB in which subscription data is stored, local CSV/JSON files in which historical dengue cases, weather data, GeoJSON maps and forecast outputs are stored. It is advisable to have periodic backup of these files .

## 4.5 API Endpoints

- **GET /api/national** – Access national-level dengue forecast data, possibly with a date range filter.
- **GET /api/provincial** – get provincial-level dengue predictions. Province name and date range can be filtered.

- **GET /api/district –** getting forecasts of dengue at the district level, optionally by district name and date range
- **POST /api/chat –** Chat with dengue recommendation chatbot and send user message to chatbot, and get model generated response back.
- **POST /api/subscribe –** Save user information (name, city, FCM token) in order to receive location-based dengue notifications.

## 4.6 Activity Diagram

An activity diagram is a workflow or progression of activities of a system. It graphically shows the interaction of the users or system components in stages, decisions, parallel processes and potential outcomes. These charts assist in getting the general behavior, logical flow and how the system reacts to various user actions. Activity diagrams can be particularly used to represent dynamic processes, including the processes of logging in, processing data, or interacting with users in web and mobile apps.

### 4.6.1 Web Dashboard Activity Diagram

The web activity diagram shows the general interactions on the web dashboard of the dengue prediction and alert system. It starts when the user opens the site and the location permission is granted. In case of refusal or lack of the high-risk area, the user will be sent to the dashboard. After being placed on the dashboard, the user is able to see dengue forecasts, research statistics, see precautionary details or talk to the chatbot to get a guide. The diagram reflects the most important interactions of the system modules in a simple and sequential manner, demonstrating how users navigate through the main characteristics of the web platform.

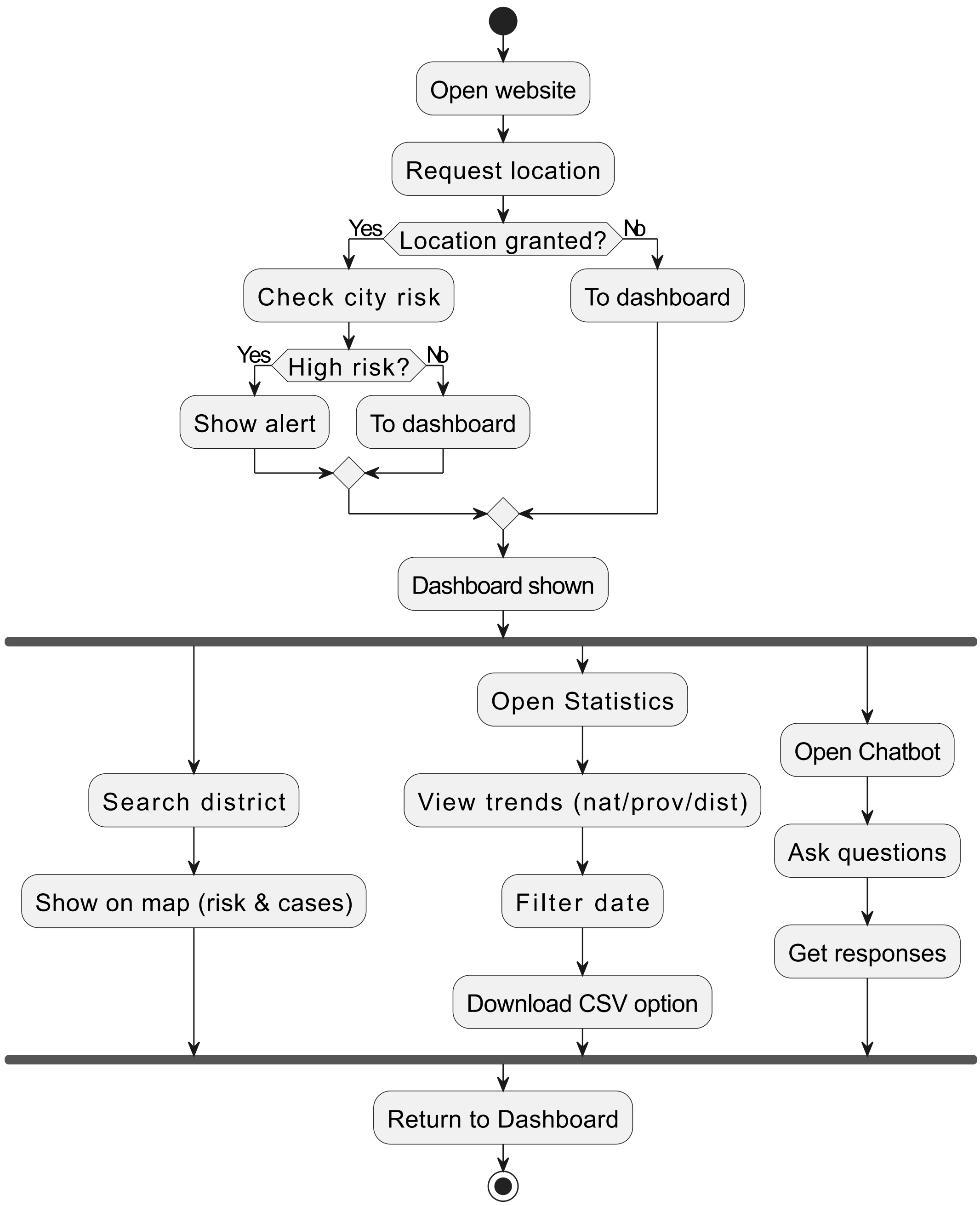


Figure 4.4: Web Activity Diagram

### 4.6.2 Mobile App Activity Diagram

mobile application activity diagram shows the user flow and it begins at the home screen, where several feature buttons are shown. The users are free to scroll to any area like the map, statistics, chatbot, precautions, or symptoms. In case the user decides to complete the alert form including his or her name and city, the system will evaluate the city that he/she has entered to see whether it is under high risk at the moment. In such a case, when the app is open, an in-app banner is displayed; if not an actual push notification is delivered to the phone. The statistics section enables the user to consider the trends of dengue cases at the national, provincial, or district level, use a date range to select the data, and save the data in a CSV file. The chatbot offers an automated response which concerns dengue and the signs and precautions section has useful informational content.

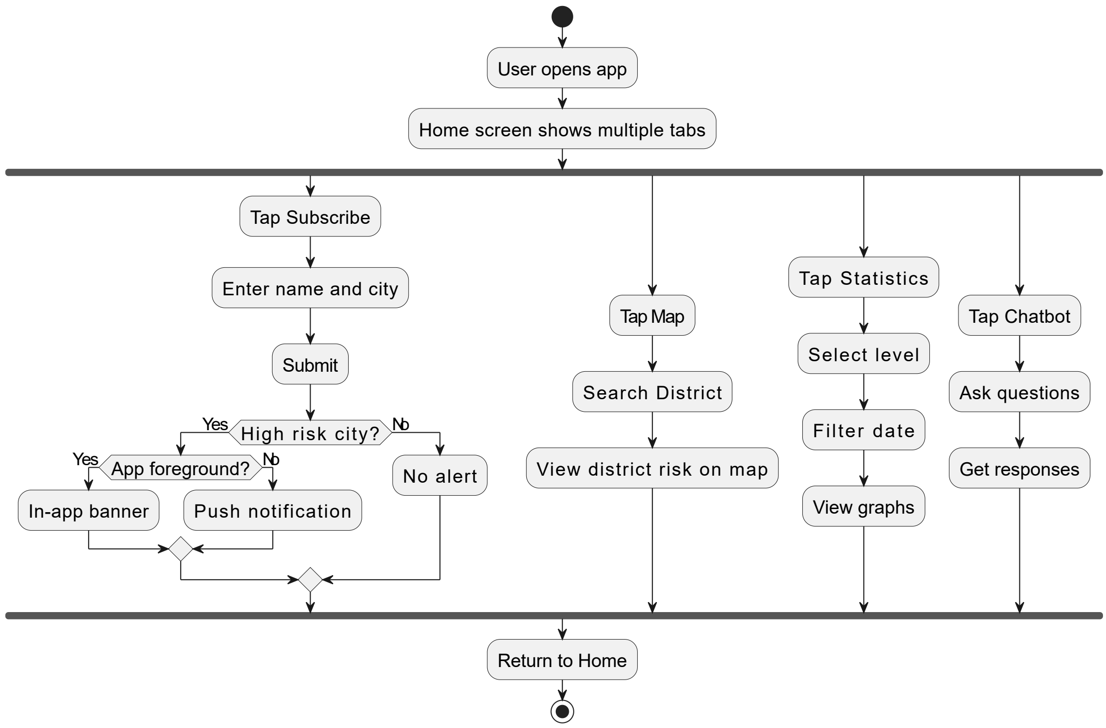


Figure 4.5: App Activity Diagram

## 4.7 Workflow Diagram

This workflow diagram describes the general operation of the Dengue Prediction and Alert System, including the interaction of the various modules that are used to present the predictions, alerts, and user support. It starts by the user opening the web or mobile dashboard. In the case of the web application, the system is asking the user to give location access, which is used to determine their location in terms of city. In case of permission, it assesses the dengue level of risk in the area. The user is alerted immediately on screen in case he is in a high-risk area to inform him about the possible danger in his area.

After loading the dashboard, the user can mingle with various features as per his or her needs. Search District option enables users to get predictions of dengue risks within a particular district by making a request to the backend and the AI model is used to analyze the information and results are given as a map with color coded risk indicators. Using the View Statistics feature, users can visualize the trends of dengue at the district, provincial, and national levels using a date filter, charts, and download CSV reports to further analyze such trends. Chatbot feature will connect the users to the smart dengue alert device that can respond to questions about dengue in real-time, using the backend and Hugging Face model.

The Alert Subscription allows mobile users to subscribe to alert messages about the risk of dengue on a daily basis and also provide them with the token of their city and devices. After they are subscribed, their information is stored to the database, and a backend cron job is used to examine the high-risk areas daily. Once the city of a subscribed user is detected as high-risk, the system will activate the Firebase Cloud Messaging (FCM) to issue push notifications to the specific device of the user. Altogether, this workflow provides a well-coordinated user-action, predictive-modeling, and automated-alerts to increase the dengue awareness and encourage the required preventive measure in a timely manner

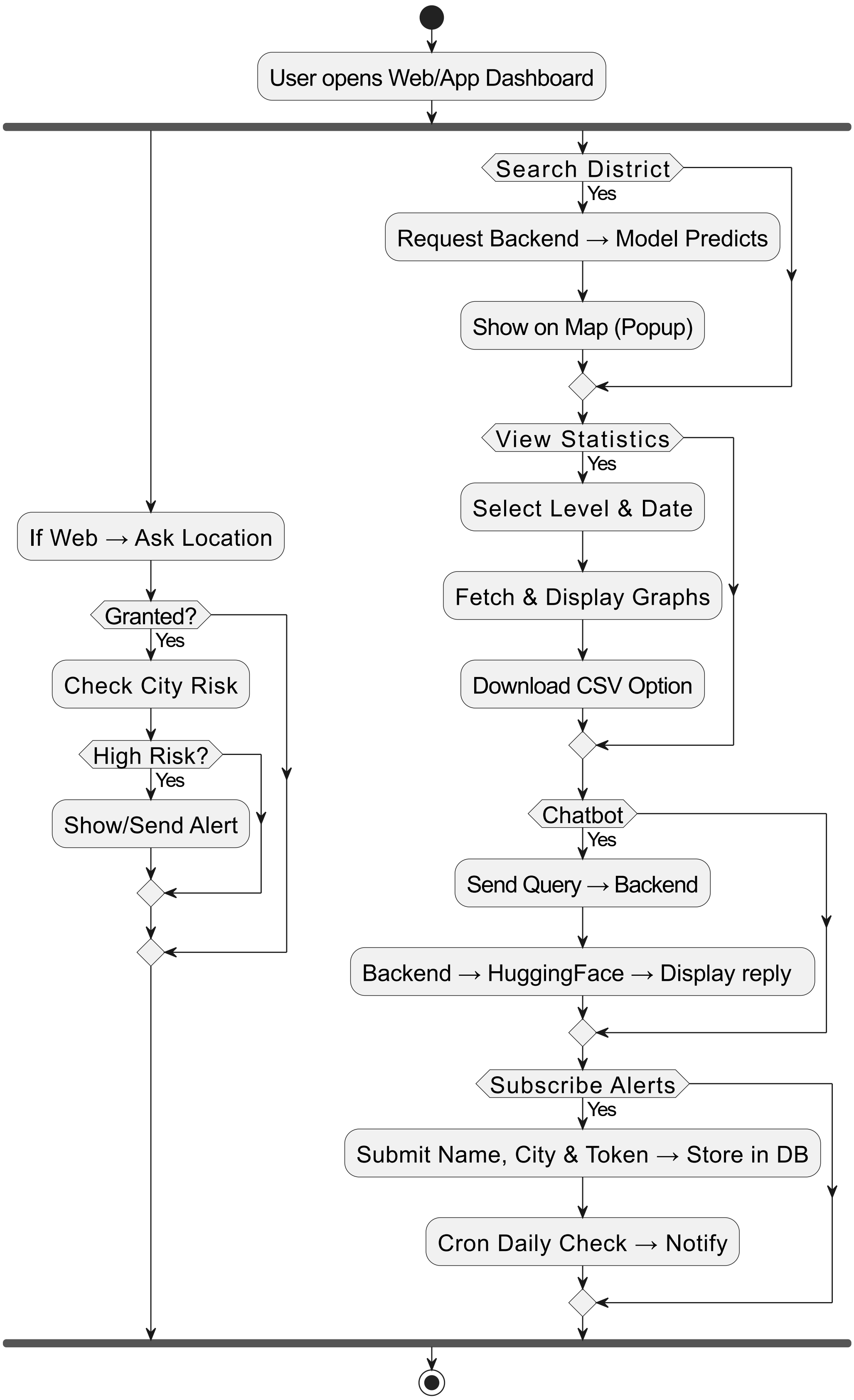




Figure 4.6: Complete Workflow Diagram

## 4.8 Low Level design

Low-Level Design (LLD) is a design phase that is concerned with the implementation of every part of a system. It subdivides the high-level modules into smaller more detailed modules specifying the interactions among components.

### 4.8.1 Sequence Diagram

A sequence diagram is a UML diagram that illustrates the interaction process between various entities of a system with time. It is concerned with the sequence of messages or actions between objects or components to achieve a given process or a given function

#### 4.8.1.1 Get Predictions Sequence Diagram

This sequence diagram shows the operation that takes place when a user has entered a particular district in the dashboard search. The typed in district is transmitted to the backend, which causes an API call to the dengue prediction model that provides predictions. The model will give the forecasted level of risk and the number of cases it is likely to have in that district. This data then forms a request to the backend which auto zooms into the district of choice on the map at the front end. A popup will be presented in the map containing the dengue risk level and the projected cases in the district.

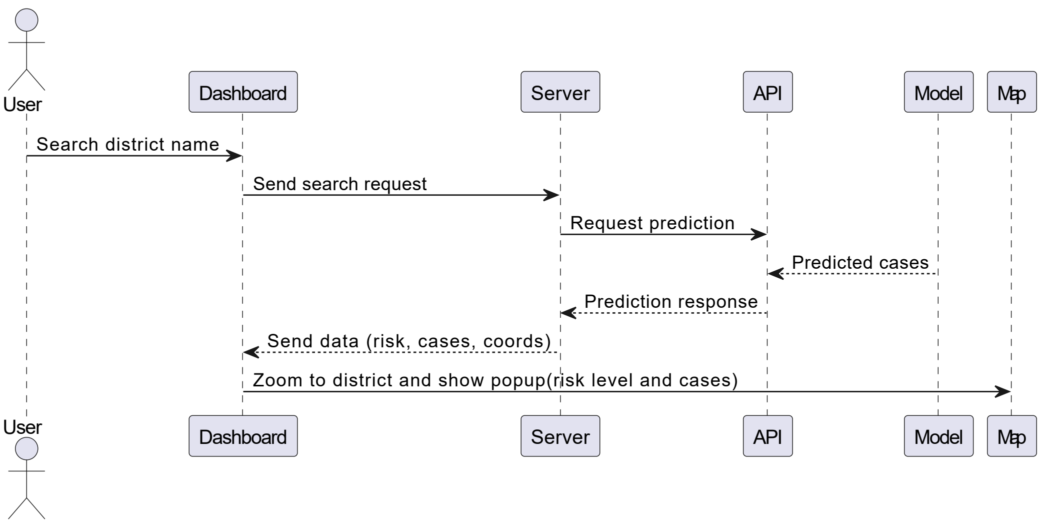


Figure 4.7: Get Predictions Sequence Diagram

#### 4.8.1.2 Alert Notification Sequence Diagram

The sequence diagram illustrates the operations of the dengue alert system of the mobile app. Upon subscription through the input of city and name the FCM token and data are transmitted to the backend that verifies the dengue risk of the city. In case the area is dangerous, Firebase Cloud Messaging triggers the notification, which can be an in-app banner when the application is open or a push notification when the application is closed so that users would be notified about dengue warnings in time.

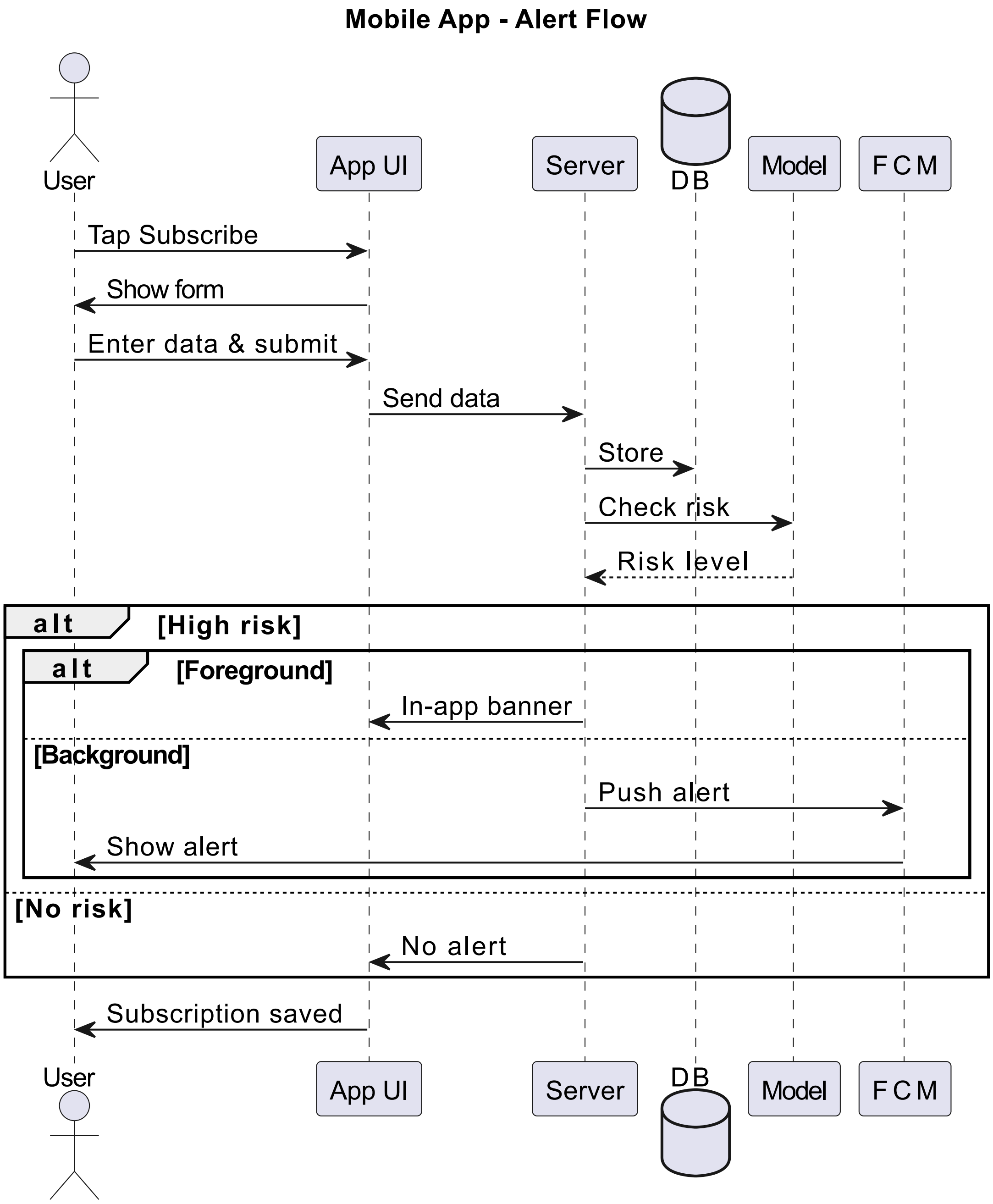


Figure 4.8: Alert Notification Sequence Diagram

#### 4.8.1.3 View Statistics Sequence Diagram

View statistics sequence diagram reveals the user interface with the statistics tab to access dengue prediction statistics. When a user clicks on the tab, he/she can choose to see national, provincial, or district-level data and filter the results based on the date range. These are passed to the backend which retrieves and processes the corresponding prediction information. The outcomes are then presented in the form of visualizations in graphs form where it is possible to see trends and comparisons. The user is also able to download the prediction data that is being viewed as an CSV file to further analyse.

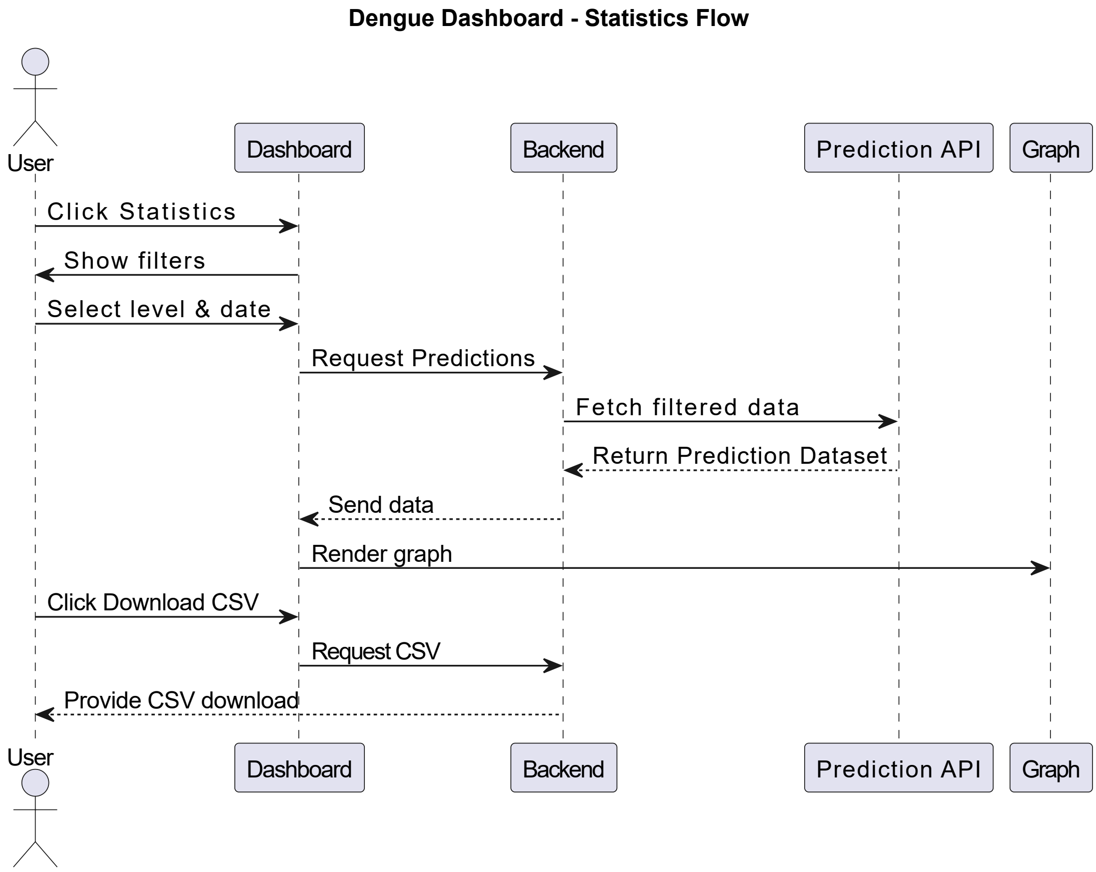


Figure 4.9: View Statistics Sequence Diagram

#### 4.8.1.4 Chatbot Sequence Diagram

The following sequence diagram shows how the dengue chatbot implemented into the system using the Hugging Face API works. Once the user makes a query by use of the chatbot interface, the query is relayed to the backend that then interacts with the Hugging Face model. The chatbot model analyses the question and formulates the relevant response that is sent back to the backend. The backend then responds with this to the frontend where it is displayed on the chatbot interface, and the user can get responses in sometime to the dengue-related questions. The system interacts with various external services to offer dynamic data-driven capabilities including weather predictive services and alert delivery.

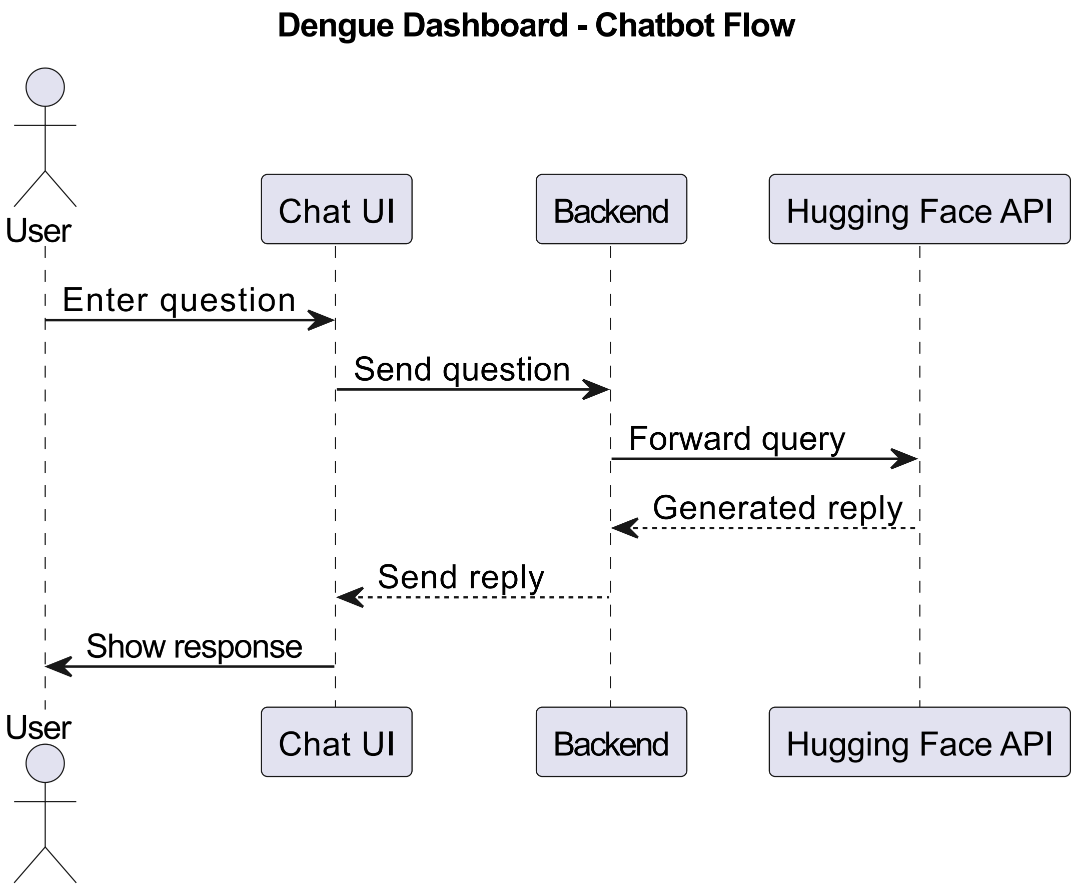


Figure 4.10: Chatbot Sequence Diagram

## 4.9 Database Design

The system uses a NoSQL or relational database (e.g., MongoDB). The database is used to store and manage user-related data and FCM token to send alerts

Table 4.1: User Data Schema

| Field Name | Type | Length | Validation Rules | Description |
|---|---|---|---|---|
| _id | ObjectId | — | Auto-generated (Unique) | Unique MongoDB document ID |
| name | String | 100 | Required | Full name of the user |
| city | String | 100 | Required | User's city for risk detection |
| Fcm device token | String | 100 | Required | Used to send push notification on device |
| createdAt | Date | — | Auto-generated | Timestamp for record creation |

### 4.9.1 Database Entity

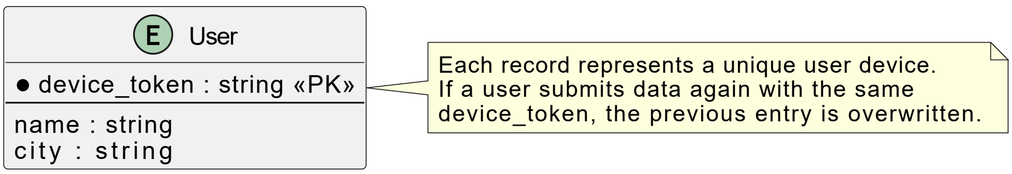


Figure 4.11: Database Entity

## 4.10 External Interface Requirements

The system interacts with various external services to offer dynamic data-driven capabilities including weather predictive services and alert delivery.

**4.7.1 External Systems**

- **Prophet Prediction Service (ML API):** Model used for dengue predictions.
- **Hugging Face / Mistral Chatbot API:**Responds to chatbots on the fine-tuned model.
- **Firebase Cloud Messaging (FCM):** It is used to give push messages and alerts regarding dengue to registered devices of the users.
- **OpenWeatherMap API:** Provides the map render and incorporates geospatial data of. pakistan.json.

- **Leaflet.js API:** Handles rendering of the map and integrates geospatial data from `pakistan.json`.
- **MongoDB:** Stores user subscriptions.

The system is based on HTTP/ HTTPS a method of communication between the front end and the backend. The main format of data is the JSON used as a request and response payload.

# Chapter 5

# System Implementation

## 5.1 System Architecture

The system architecture of the Dengue Outbreak Prediction and Warning System, a solid understanding of the interactions between various components of the system how they interact and perform. The architecture is further subdivided into various layers at the topmost level, which are data input, prediction and processing, application and backend services, data visualization and reporting, and user interaction and alerting. The data input layer gathers historical dengue case data, weather information (temperature, rainfall, humidity), and geographic boundaries (GeoJSON). These datasets are processed and sent to the Prophet machine learning model, implemented in Python, which forecasts dengue risk levels. The trained model generates prediction results locally, which are stored in a JSON file and then served to the frontend through RESTful APIs built with Node.js and Express.js .

The backend layer acts as the system's central hub, handling all API communication between the frontend, ML model, and external services. It uses MongoDB for user and token storage. Cron jobs are integrated solely to trigger Firebase Cloud Messaging (FCM) alerts for subscribed mobile users whenever their location is detected as high-risk and is also used for running python ML script. For the web dashboard, notifications are handled through custom frontend logic that checks live risk zones. On the frontend, the React.js dashboard and React Native (Expo) mobile app fetch prediction data through API calls and visualize it using Leaflet.js for maps and Chart.js for analytics. The chatbot, powered by the Hugging Face API via the Mistral model, the backend is used to process user queries that is the one sending the replies back to the UI. The use of secure HTTPS communication guarantees protected data transfer between all levels, and creating a unified and effective dengue ecosystem, visualization, alert delivery and prediction. As shown in figure 4.1

## 5.2 Tools and Technologies

### 5.2.1 Frontend

- **React.js:** It was used to develop the web dashboard, which is interactive and dynamic user interface to visualize predictions of dengue risks, obtain statistical charts and interact with the chatbot.
- **React Native (Expo):** It is used to create the cross-platform mobile app, enabling real-time notifications and access to predictions and alerts via mobile.
- **Leaflet.js:** This is an open-source Java script library that is used to render the geospatial dengue risk map, which is a map of the risk zones in color-coded districts.
- **Tailwind CSS:**It is a framework that is employed to style the frontend interfaces in a clean and responsive manner for uniform user experience at the devices.
- **Axios:** This allows communication between the frontend and the backend using RESTful API calls guarantee a seamless data flow and updates.

### 5.2.2 Backend

- **Node.js:** The main backend runtime, which receives API requests, and interface between the model server and database and the frontend interfaces.
- **Express.js:** This is a lightweight framework based on Node.js, which develops RESTful endpoints and routing administration of frontend and mobile applications.
- **Firebase Cloud Messaging (FCM):** It is used to push new notifications enabling the system to deliver dengue risk notification straight to subscribed mobile application.
- **Cron Jobs:**Schedules of routinely running dengue checks risk updates and automatic sending of notifications to registered users.

### 5.2.3 Machine Learning

The machine learning part is the analytical and predictive foundation of the Dengue Forecasting and warning mechanism. It allows proper forecasting of dengue outbreak and classification of risks at the various geographical levels. The most important technologies include:

- **Prophet Model:** A Meta time series forecasting model that is utilized to predict monthly cases of dengue at the provincial level according to previous cases and seasonal patterns.

- **Weighted Disaggregation Method:**Use of this method to assign provincial forecasts to district levels based on social-environmental variables like population, density, rainfall, and Human Development Index (HDI).
- **Python (Pandas, NumPy, Scikit-learn):** Preprocessing, feature selection and statistical calculation then inputs data into Prophet model.
- **REST API Integration:** The trained Prophet model is deployed independently and served via an API, to enable the backend to retrieve predictions on demand to display.

### 5.2.4 Database

- **MongoDB Atlas:** Data is stored in the form of structured data, e.g. user subscriptions and FCM tokens provisioning of scalable and flexible data management.

### 5.2.5 Cloud and Integration Services

- **Firebase Cloud Messaging (FCM):** Pushes mobile with real-time notifications to users in case of high risk dengue alert.
- **Hugging Face (Mistral Model):** Implemented to run the chatbot feature to allow natural language processing to support dengue queries.

### 5.2.6 Data Visualization and Reporting

- **Chart.js:** it will be used to create dynamic line, bar, and pie charts that will show dengue sleepless tendencies and statistics.
- **Leaflet.js:** Displays interactive maps of the dengue risk areas within the district in color coded indicators.
- **CSV Export Utility:**This enables users to export dengue case statistics and prediction information to be further analyzed or recorded.

| Tool | Used For |
|---|---|
| **Overleaf** | Project documentation and report writing in LaTeX format. |
| **Visual Studio Code** | Coding and integration of frontend (React.js, React Native) and backend (Node.js, Express). |
| **Draw.io** | Designing use case diagrams, activity diagrams, and flowcharts. |
| **Google Colab** | Training and testing the Prophet machine learning model and Mistral chatbot. |
| **Hugging Face** | Deploying and integrating the Mistral chatbot through API. |
| **Firebase** | Sending mobile push notifications to subscribed users using FCM. |
| **MongoDB** | Storing user details, tokens, and managing backend data structure. |

Table 5.1: Tools Used in System Implementation

### 5.2.7 Summary of Implementation

Dengue Prediction and Alert System is a system that combines several technologies on the frontend, backend and machine learning layers to develop a single, smart platform of public health awareness. Its scalability, reliability and future ease of use are achieved through its modular architecture scaling to deliver more predictive analytics and user interaction.

## 5.3 Methodology

The creation of the system of DenguePredict created through a systematic and repetitive process in order to achieve the necessary precision, functionality, and compatibility of the different elements. The methodology explains the system design, implementation and testing process of the system beginning with the data collection process to the final development of the system.

### 5.3.1 Step 1: Data Collection and Preprocessing

- Historical records of dengue cases were taken from Open Dengue and Kaggle.
- External APIs were used to gather weather variables (rainfall, temperature, humidity) and they were stored locally in CSV format..
- The GeoJSON file of Pakistan was used to visualize the map by using the geographical borders of Pakistan in the map visualization.
- The datasets were cleaned, the missing values dealt with, and formatted to resemble the input format of Prophet model and removing some inconsistencies.

### 5.3.2 Step 2: Prediction Model Development (Prophet)

Prophet was a time series forecasting model applied because it has a high quality on seasonal data..

- The model was trained using historical data of dengue and weather variables to predict monthly dengue[21].
- The test and evaluation of the model was done using cross validation method until credible forecasts were achieved.
- The estimated cases monthly were then assigned to the districts through the multifactor proportional allocation technique by capitalizing on variables like:
  - Average rainfall
  - Human Development Index (HDI)
  - Population density
  - Total population
- Predictions that were made were saved to be used by the backend API and presented to the web and mobile interfaces.

### 5.3.3 Step 3: Backend Development (Node.js)

A Node.js server was created to handle all the logic in the system and provide information to the applications.

- The endpoints of API were developed for:
  - National, provincial, and district-level predictions
  - User subscriptions (saving device tokens)

  - Chatbot forwarding (forwarding requests to HuggingFace)

- The introduction of cron jobs was used to send daily alerts automatically on Firebase Cloud Messaging (FCM) depending upon the most recent forecasted risk level.

- The user subscription data such name, city, and the device token were stored in MongoDB with the following data.

### 5.3.4 Step 4: Frontend Development (Web Dashboard)

A React.js web dashboard was designed to display predictions in a graphic manner.

- Significant UI elements were:
  - GeoJSON interactive district-level map.
  - District look up search bar.
  - Real time national, provincial and district data and download csv.
  - Weather display
  - Symptoms, precautions, and the chatbot pages.

- Background APIs were incorporated with all the components to display live results.

### 5.3.5 Step 5: Mobile App Development (React Native Expo)

The system was made available to more people through the development of a mobile application.

- Features included:
  - Know Your Risk Interactive Map.
  - Chatbot interface
  - Statistics visualizations
  - Symptoms and precautions pages
  - Dengue alerts sub-form.

- The app shares the same backend APIs as web dashboard.

### 5.3.6 Step 6: Notification System Using Firebase

The web and the mobile application were both integrated with Firebase Cloud messaging (FCM).

- Every subscribed member is entitled to:

  - Notifications about the forecasted danger of dengue in their city every day.
  - Cron job notification messages.

- MongoDB holds device tokens in a secure place.

### 5.3.7 Step 7: Testing and Integration

All modules (backend, model, UI, notifications) were tested separately.

- Testing ensured that:
  - Predictions flowed correctly from Prophet to the backend and then to the frontend
  - Notifications are activated properly.
- The iterative process was done to debug and make improvements to the process to achieve more and more quality and better predictions.

## 5.4 Experimental Work

### 5.4.1 Objective

The purpose of the experimental work consists in the measurement of accuracy, reliability as well as predictive performance of the dengue forecasting system. The experiments are concerned with the integration of epidemiological information and climatic factors including temperature, rainfall and humidity. The Primary objective is to evaluate the ability of Prophet to model the temporal and season- trends in the provinces and districts of Pakistan. Also, the analysis considers how weather regressors enhance prediction accuracy and the issue of regional differences predictions of dengue outbreaks.

### 5.4.2 Dataset

The collections of data were gathered in a number of open sources:

- **Dengue Cases and Deaths:**OpenDengue and Kaggle collected data, which includes monthly province-wise cases and fatalities of dengue.
- **Weather Data:**Fetching the Open-Meteo API, with monthly averages of temperature, humidity and rainfall of every province.
- **District-Level Factors:**To be included, they are to be compiled manually using secondary sources included in socio-demographic and environmental indicators are population, population density, HDI and average rain fall, and then plotted to the provinces.

### 5.4.3 Data Preprocessing

In order to get the datasets ready to be used in forecasting, the steps listed below were used:

1. **Cleaning and Normalization:** The missing values were eliminated, and inconsistent values were fixed.
2. **Mapping:** Mapped dengue regions to corresponding provinces for weather integration (e.g., “KPK” → “Khyber Pakhtunkhwa”).
3. **Date Conversion:** Converted monthly and yearly data to good datetime objects for Prophet modeling.
4. **Merging:** Adding dengue and weather data on a monthly and provincial basis to create unified training data.
5. **Feature Engineering:** Prophet has been introduced to features (temperature, rainfall) apprehend the environmental influence.
6. **Missing Data Handling:**Forward/backward filling and climatology-based replacement of missing weather records.
7. **Validation:** Checked all convertions of numbers and ensured consistency of before model training temporal ordering.

### 5.4.4 Model Composition

Dengue forecasting model was applied on the Prophet library because of its reliability in the seasonality and changes in trends. The system architecture is comprised of the following modules:

- **Prophet Core Model:** The model is set with annual seasonality and monsoon months as seasonal outbreaks to be caused by custom holiday effects.
- **Weather Regressors:** Temperature, rainfall and humidity were added as additional regressors to model environmental effect.
- **Regional Forecasting:** Prophet Independent models were trained province wisely to model localized patterns, where population and adjustments at the district level are used environmental indicators.
- **Forecast Extension:** The model made monthly forecasts, which generated the mean, upper, and lower confidence levels of each province.

### 5.4.5 Evaluation Metrics

Model validation was performed using rolling-origin time-series cross-validation with a 6-month forecast horizon for each administrative region, ensuring no future data leakage.

- **Mean Absolute Error (MAE):** Measures the deviation between actual and predicted dengue cases.
- **Mean Absolute Percentage Error (MAPE):**Expresses the prediction error as a percentage of the actual values.
- **Root Mean Square Error (RMSE):**A scale-dependent error metric that gives higher weight to larger errors.

### 5.4.6 Results

- The model achieved consistent forecasting accuracy with an average MAPE below 5% and the average MAE 849.
- Integration of weather regressors significantly enhanced prediction stability, especially during monsoon months.
- The average prediction generation time per province was approximately 1.5 seconds.
- Province-wise forecasts were exported in JSON format for visualization in the dashboard.

The experimental results demonstrated that combining epidemiological data with climatic regressors substantially improved dengue outbreak prediction accuracy. Prophet's inclusion of monsoon-based seasonality captured outbreak peaks effectively, and preprocessing ensured consistent provincial and district-level forecasting. The resulting forecasts were successfully visualized on the web dashboard for actionable dengue surveillance.

## 5.5 User Interface Snapshots

### 5.5.1 Web Interface

#### 5.5.1.1 Web Dashboard

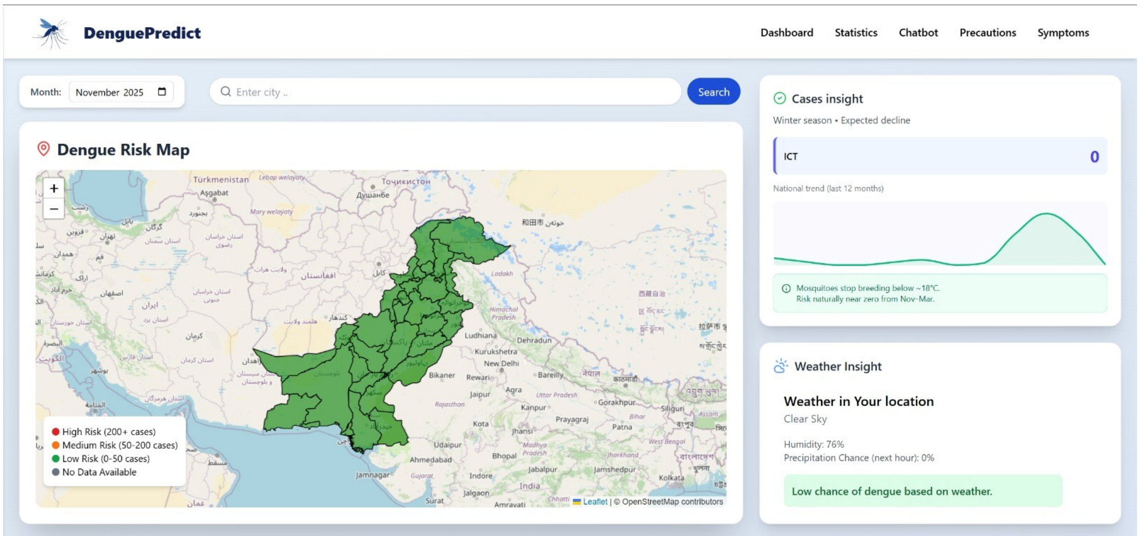


Figure 5.1: Web Dashboard

The DenguePredict Web Dashboard has a fast access to main pages, such as Symptoms, Precaution, Chatbot, and Statistics, as well as a search bar to verify dengue risk in districts. The user is able to pick a month or a year, see an interactive district map with risk pop-ups, and view live cases including weather conditions on the right hand side. There is a footer and navbar for consistent navigation across pages.

### 5.5.1.2 Statistics Page


Figure 5.2: Statistics Page

The Statistics Page enables the user to browse the dengue statistics either at the National or Provincial or District level has a date range picker. It shows forecasts and graphs in form of visuals depending on the predictions selected filters. The Provincial and district level is chosen in the presented figure, which offers the details dengue trends by the individual province

## 5.5.2 Mobile Application Interface

### 5.5.2.1 Dashboard

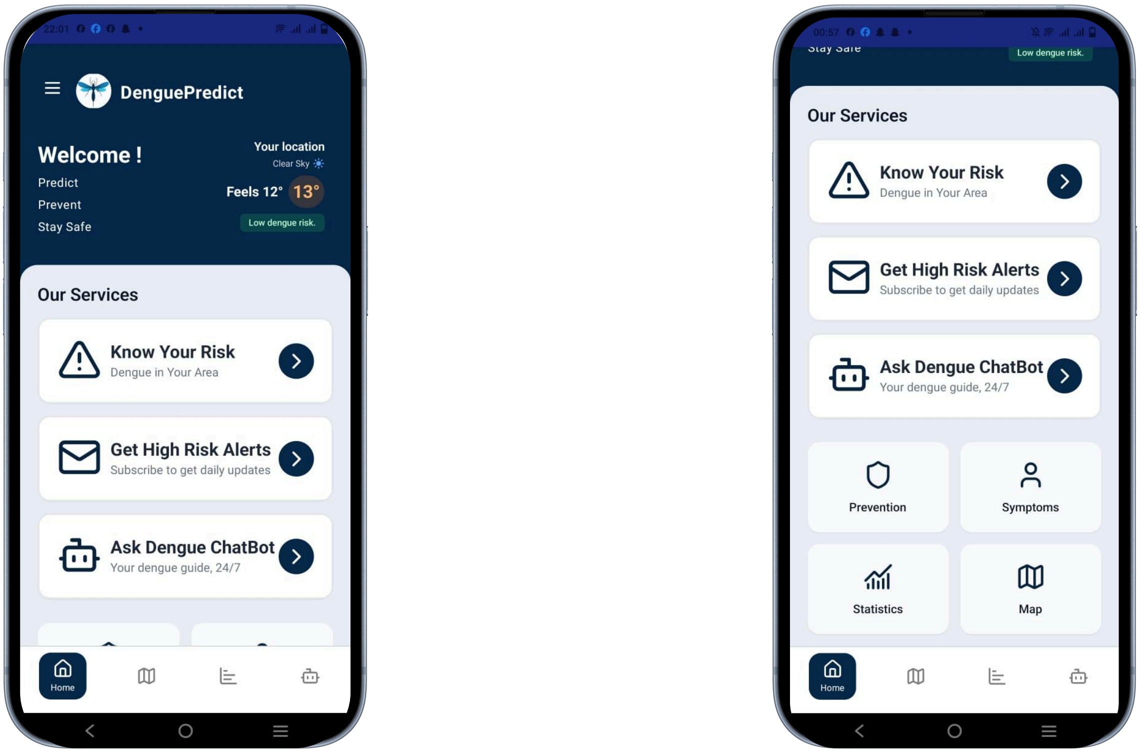


Figure 5.3: App Dashboard

The application has a user-friendly dashboard and integrates buttons to Know Your Risk, Chatbot, Alerts Subscriptions, Symptoms, Preventions and Statistics. It has a side navigation Drawer as well giving easy access to all the key features that make user access easy and smooth navigation. Every button or drawers feature takes users to the corresponding feature page ensuring seamless interaction.

### 5.5.2.2 Chatbot Interface

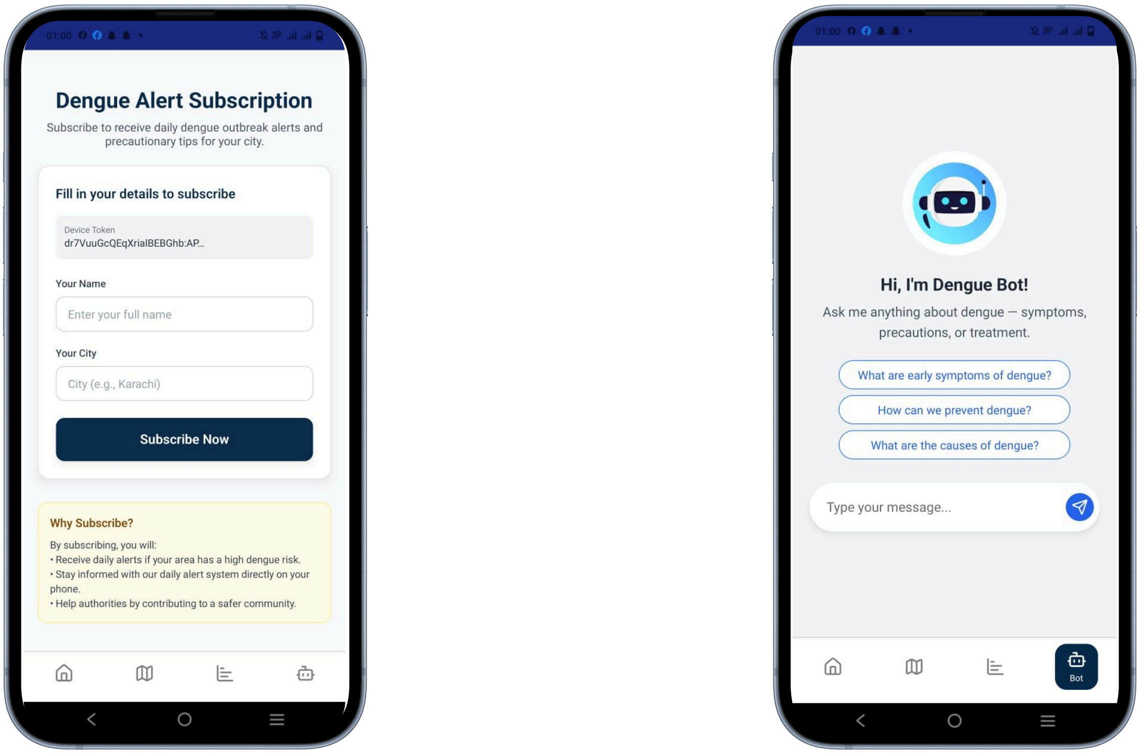


Figure 5.4: Chatbot and Alert Subscription page

Chatbot Page is a user-friendly site with specific quick questions on issues as well as early symptoms and precautions, so that people can have access to immediate instructions.

### 5.5.2.3 Alert Subscription

The Subscribe for Alerts Page has a simple form displaying important user information (Name and City) to fall under smooth registration. When submitted the information of the user stored is safely in the database where they are automatically subscribed to the high risk alert on a daily basis. shown in figure above 5.4

# Chapter 6

# System Testing and Evaluation

## 6.1 System Overview

Dengue Outbreak Prediction and Warning System incorporates a number of important elements,also outbreak forecasting implemented with machine learning, visualization, location-based notifications, chatbot support, and cross-platform applications (internet and mobile). Owing to its thorough testing and evaluation, modular and data-driven architecture was necessary.

The web and mobile dashboards, backend APIs were tested built on Node.js, the Prophet-based prediction model, Firebase notification services, and. chatbot logic combined with Hugging Face API. There was also end-to-end testing that was conducted to authenticate user interactions like viewing risk maps, receive alert, and conversing with the chatbot. The major concern was to verify the accuracy of dengue forecasts, dashboards reaction, dengue predictability and alert mechanisms.

## 6.2 Test Approach

The test plan that was used in this system was designed into various phases to test the separate parts as well as their interactions as a coherent whole. The approach unit testing, integration testing, system testing and other.

### 6.2.1 Unit Testing

The individually correctness of the components was tested by doing unit test to each component. In the case of the frontend, the tests were to be conducted on map rendering, alert notifications, and chatbot message, processing, and visualization of data charts. Individual API endpoints found in the backend such as /subscribe, and /district were also tested with mock data to validate request and response formats. The correct forecast of the Prophet model functions was also checked. generation and date alignment.

### 6.2.2 Integration Testing

Integration testing had to make sure that communication between various modules frontend, backend, Machine learning model and database worked perfectly. Tests validated data flow between the Prophet and the backend, the prediction synchronization of the district level using the React dashboard, and between Firebase notifications and the subscribed user database. The fact that prediction data retrieved was successful was confirmed by test cases the visualization of the model was done correctly and alerts were raised automatically when high-risk was detected.

### 6.2.3 System Testing

System testing was used to test the platform as a whole within real-world conditions. Both the web and mobile applications were subjected to user behavior test like allowing access to location, predicting risk rates, communicating with the chatbot, and subscribing for notifications. The system was tested to be properly managing user permissions correctly at maps, and interactive movement between dashboard modules.

## 6.3 Unit Testing

### 6.3.1 Frontend (React.js Dashboard)

Frontend testing was aimed at making sure all interface elements rendered and acted correctly. Tests were conducted on map loading, selection of the district, data retrieval and display of alerts. Instead, the system was tested to be responsive to a variety of screen sizes browsers. The input validation was done to ensure the user location permissions were handled. The color-coded map markers were tested to ensure that it was properly represented as low, medium and and high-risk zones. The browser notification mechanism was tested to ensure that it requested proper permission at the right time and provided alerts at the right time when the user is at the right position coordinates was associated with a high-risk region.

### 6.3.2 Backend (Node.js and Python APIs)

The quality of data communication between the machine was validated using backend unit tests learning prediction model and the frontend interface. Testing of the Node.js API routes was done to correct responses, good error treatment and valid output forms. The Python prediction model was exploited alone to evaluate the fact that it generated real-time dengue risk forecasts using environmental and weather conditions. Mock data tests were used to make sure that invalid or missing inputs produced the right error message without system crashes.Firebase was combined with Alert logic and was tested to work sending of messages and recovery in case of failure in delivery.

## 6.4 Integration Testing

Integration testing was done to ensure that there is smooth communication among all core modules, so that data was flowed in the correct direction.

### 6.4.1 Test Case: Search District Risk

Table 6.1: Test Case for Search District Risk

| | |
|---|---|
| **Test Case ID:** | TC1 |
| **Description:** | Verify that users can search any district name to view its dengue risk level and predictions retrieved from the backend API. |
| **Setup:** | The AI model must have district-wise dengue risk data. The frontend and backend connection should be active. |
| **Input:** | Enter district name (e.g., “Lahore”) in the search bar. |
| **Instructions:** | 1. Open the dashboard.<br>2. Enter valid district name in type a search box.<br>3. Click the search icon.<br>4. See the presented estimated cases and Risk level. |
| **Expected Result:** | * The system retrieves and zooms map on that district with a pop up showing dengue risk level (High/Medium/Low) and Predicted Cases. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test case confirmed the collaboration between the frontend search module and the backend prediction API. It was necessary in order to ensure that a user typed the name of a district and the system would get it right fetched and showcased the dengue risk level and anticipated cases on the map. Successful performance verified the correctness of API communication, proper map display, and reliable inter-module prediction data synchronization.

### 6.4.2 Test Case: View Risk Map

Table 6.2: Test Case for View Risk Map

| | |
|---|---|
| **Test Case ID:** | TC2 |
| **Description:** | Verify that the interactive map loads successfully and shows color-coded districts representing dengue risk levels. |
| **Setup:** | System connected to prediction API to show risk data.System should load pakistan.json file to show map coordinates. Internet connection available. |
| **Input:** | Open the map tab on the dashboard. |
| **Instructions:** | 1. Click on the “Map” tab.<br>2. Wait for the map to load.<br>3. Note colors of risk levels of districts.<br>4. Click on a district to access the popup. |
| **Expected Result:** | * Map loads correctly.<br>• Districts display appropriate colors: red (high), yellow (medium), green (low).<br>• Popup shows predicted cases and risk level. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test ensured the correct interaction between the map visualization component, Pakistan.json and backend prediction outputs. It verified that color-coded regions accurately represented dengue risk levels and that clicking on any district displayed its detailed popup. The test confirmed map stability, responsiveness, and accurate visual mapping of dynamic data.

### 6.4.3 Test Case: Receive Alerts (Push Notifications)

Table 6.3: Test Case for Receive Alerts (Push Notifications)

| | |
|---|---|
| **Test Case ID:** | TC3 |
| **Description:** | Verify that users receive dengue alerts via Firebase Cloud Messaging when their subscribed city is at high-risk. |
| **Setup:** | Firebase Cloud Messaging configured; user has submitted subscription form with name and city. Backend alert module active. |
| **Input:** | Subscribe with user details (name, city). |
| **Instructions:** | 1. Open the "Alerts Subscription" form.<br>2. Enter name and city.<br>3. Click Submit.<br>4. Wait for the system to detect a high-risk update for that city.<br>5. Observe for push notification |
| **Expected Result:** | * User is subscribed successfully.<br>• Notification appears if city enters high-risk condition.<br>• Message includes city name and risk level. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test case evaluated the integration between the Firebase Cloud Messaging service, backend alert module, and user subscription form. It ensured that subscribed users received timely notifications when their selected city entered a high-risk condition. The test confirmed proper backend-to-Firebase communication and successful delivery of push notifications to users.

### 6.4.4 Test Case: Chatbot Support

Table 6.4: Test Case for Chatbot Support

| | |
|---|---|
| **Test Case ID:** | TC4 |
| **Description:** | Verify that the chatbot responds correctly to dengue-related user queries. |
| **Setup:** | Chatbot API (Mistral model) must be active and connected to the frontend. Internet available. |
| **Input:** | Type: "What are symptoms of dengue?" |
| **Instructions:** | 1. Open chatbot tab.<br>2. Wait for interface to load.<br>3. Enter dengue-related question.<br>4. Press Enter.<br>5. Observe chatbot reply. |
| **Expected Result:** | * Chatbot responds with accurate and relevant information.<br>• Response is displayed in real-time chat format. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test verified the communication flow between the frontend chatbot interface and the Mistral-based chatbot API. It checked whether the chatbot provided accurate, context aware responses to dengue-related queries. The test confirmed that natural language queries were processed correctly and that the responses appeared in real-time chat format with consistent UI feedback

### 6.4.5 Test Case: View Statistics

Table 6.5: Test Case for View Statistics

| | |
|---|---|
| **Test Case ID:** | TC5 |
| **Description:** | Verify that users can view dengue trends and statistics through interactive graphs and download the displayed data. |
| **Setup:** | System must have dengue historical and predicted data available. Graph and chart rendering modules active. |
| **Input:** | Open Statistics page and select date range. |
| **Instructions:** | 1. Click on "Statistics" tab.<br>2. Wait for graphs to load.<br>3. Choose desired date range.<br>4. Observe graphs update dynamically.<br>5. Click Download CSV. |
| **Expected Result:** | * Graphs display dengue trends accurately.<br>• Downloaded CSV contains prediction data.<br>• System remains responsive during updates. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test validated integration among the data visualization module, prediction database, and frontend dashboard. It ensured that dengue trends were displayed accurately in dynamic charts and that users could download data as CSV files. The system's responsiveness during graph updates and data exports was also verified, confirming smooth user interaction and reliable data handling.

### 6.4.6 Test Case: Read Awareness and Guidance Info

Table 6.6: Test Case for Read Awareness and Guidance Info

| | |
|---|---|
| **Test Case ID:** | TC6 |
| **Description:** | Verify that users can access and read dengue-related awareness information from the "Symptoms" and "Precautions" sections. |
| **Setup:** | System must have added Symptoms and Precautions information. |
| **Input:** | Click "Symptoms" tab, then "Precautions" tab. |
| **Instructions:** | 1. Click the "Symptoms" tab.<br>2. Observe the content displayed.<br>3. Click the "Precautions" tab.<br>4. Verify that preventive steps appear. |
| **Expected Result:** | * Awareness content loads correctly.<br>• Text is readable and informative.<br>• Proper error message shown if content unavailable. |
| **Actual Result:** | As expected. |
| **Verdict:** | Pass |

This test case assessed the accessibility and functionality of the awareness content sections ("Symptoms" and "Precautions"). It ensured that health guidance information loaded correctly from the backend and displayed neatly on the frontend. The test confirmed proper content management, error handling, and readability of educational material for end users.

## 6.5 Summary of Testing and Evaluation

In summary, the Dengue Outbreak Prediction and Warning System underwent rigorous testing across all layers of its architecture. Unit and integration tests ensured that individual components operated correctly and interacted smoothly. System testing verified real-world usability and reliability across web and mobile platforms. White box tests confirmed both the correctness of internal logic and the accuracy of userfacing behavior. Overall, the system achieved high stability, precise prediction accuracy, and responsive alert delivery successfully meeting the intended goals of awareness, prevention, and public health support.

# Chapter 7

# Conclusions

The **Dengue Outbreak Prediction and Warning System** has been created in the intention to bring an intelligent and data-driven platform of predicting the possible dengue outbreaks and delivering a timely alert to the users. The system has achieved its goal of illustrating the way data science could be deployed to public health management, accessible and potentially useful, through the combination of machine learning, geospatial visualization, and real-time communication technology.

The project implementation has also given me important experience of end-to-end process of creating an AI-based web and mobile system, including data collection and preprocessing, model deployment, and real-time user interaction. Prophet model made time-series forecasting efficient, whereas Firebase and browser notifications integration made sure that the user was informed about the dengue-related alerts, based on his/her location. Additionally, the interactive dashboards developed on the basis of React.js and React Native resulted in the improvement of user experience because risk levels and statistical trends as well as the results of prediction could be properly visualized

## 7.1 Key Contributions

The creation of the Dengue Prediction and Alert System has contributed significantly to the advancement of the attitudes of people to the health of their bodies, prevention of the disease, and informed decision making. The system incorporates technology and health analytics to deliver timely and actionable insights to the communities and health authorities. The significant contributions of this work are:

- **Public Health Empowerment:** The project will improve health data to public awareness through the provision of early dengue risk forecasting and ready precautionary health advice to users in dengue-endemic locations.

- **AI-Driven Prevention:** The system is capable of data-driven forecasting using the model of machine learning to reduce the severity of outbreaks and assist in the better prevention.
- **Accessible Alert Mechanism:** In both web and mobile interfaces, users are to be notified in time and be able to monitor the risk of dengue.
- **Community Awareness Enhancement:** The project fosters the proactive behaviour of citizens by providing localised alerts, preventive tips and educating information on dengue prevention.
- **Scalable Technological Foundation:** It is also scalable to other infectious disease monitoring systems, which is made affordable by the modular and data-centric nature of the system which enables easy integration of future features in the system.

### Achievements

This system achieves multiple objectives:

- Provides early warning by predicting dengue risk at provincial and district levels.
- Enables visual risk mapping and trend analysis using interactive charts and maps.
- Delivers automated, location-based notifications to keep users informed.
- Integrates a conversational chatbot powered by Hugging Face for dengue-related queries.
- Offers both web and mobile access, ensuring wide usability and reach.

## 7.2 Challenges Faced

Some of the challenges that were experienced during the development were in the data, technical, and operational fronts:

- Data Quality and Availability: The accuracy and district-level dengue and weather data was not easily collected because there was inconsistency in open-source data and lack of records.
- Accuracy of the Model: The prediction accuracy and real-time performance had to be balanced through the careful selection of the model and hyperparameters.
- Complexity of Integration: Co-ordination of various technologies FastAPI backend React frontend, Firebase notifications and postgreSQL database accumulated required careful co-ordination and debugging.

- Geospatial Handling: To get a good and correct map polygons of all districts in Pakistan,
- Notification Testing: Dynamism of real-time notifications and push notifications based on different user events was a problem because of the time-based notification and the reliance on the running services.

These difficulties notwithstanding, the gradual testing, validation and optimization delivered a stable and efficient system.

## 7.3 Limitations

Although successful, there are still certain constraints. The system is based on the quality and availability of public health datasets which in some cases are incomplete and late. The accuracy of the prediction model may differ with the promptness and regularity of the data updates and because of the lack of the ability to recognize the anomalies. Moreover, alerts are efficient in apps, but SMS or more extensive public alerts integration might improve further availability.

## 7.4 Future Improvements

This system can be improved in the following directions in the future.

### 7.4.1 Additional Features

Although the system is efficient in terms of predicting dengue outbreaks and providing timely alerts, developmental changes in the future can help improve its predictive and practical functions:

- **Report Analysis Module:**Incorporate a diagnostics option in the chatbot whereby users can upload reports of cases of dengue or lab results to be deciphered by the AI tool to provide individual suggestions.
- **Image-Based Diagnosis:**Use a model of vision identification to detect mosquito species or skin symptoms when uploaded photos are used to support earlier detection.
- **Feedback and Reporting System:** Add user feedback and disease-reporting options to crowd source dengue activity updates.
- **Chatbot Enhancement:** Integrate with the prediction module that will be giving personal precautionary advice depending on the users city, the predicted level of risk, or the trends of dengue outbreaks in the region.

### 7.4.2 Extended Platforms

To reach and be useful to the maximum, the system can be extended into additional health related and public platforms instead of new versions of the app:

- **Integration with Health Systems:** Altered to government or NGO health portal to have dengue surveillance and awareness campaigns synchronized.
- **IoT Sensor Connectivity:** Co-operate with intelligent environmental sensors to monitor the mosquito density and weather conditions in real- time.
- **Public Information Displays:**Post dengue alert modules on the municipal websites and public digital boards to air the risk information.
- **Cross-Disease Adaptability:** Individuals can use similar modeling strategies to forecast and control other diseases spread by vectors like malaria.

## 7.5 Final Thoughts

The Dengue Prediction and Alert System proves that technology can play an important role in ensuring the establishment of a healthier and safer community. The project combines predictive modeling, geospatial analytics, and real-time notifications, which can help focus on a significant public health issue with an innovative, information-driven solution. It does not only create awareness but also enables people and organizations to act in a preventative manner before the outbreaks escalate. It is the system that can be expanded into an intelligent surveillance of the disease in Pakistan and other countries. It can be a multi-purpose health monitoring ecosystem in the future, with the addition of diagnostic modules and integration with the IoT.

This project is a significant move to integrate artificial intelligence, social responsibility, and public health to have a stronger and more informed society.

## Summary

The project successfully demonstrates how artificial intelligence and web technologies can be used to tackle real world health issues. The system supports public awareness as well as serves as a valuable resource for researchers and health authorities.

# Appendix A

# User Manual

The user manual for DenguePredict web app and mobile application is as follows:

## A.1 Web Application

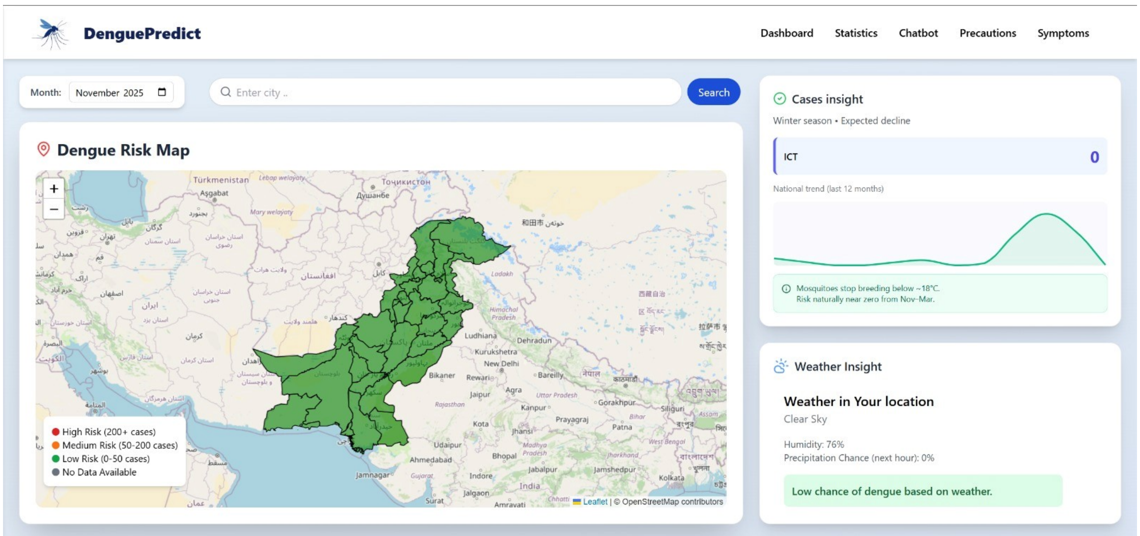


Figure A.1: Web Dashboard

### A.1.1 Main Dashboard

#### A.1.1.1 Navigation Bar

The main navigation bar consists of the following tabs for quick access:

- Home(Dashboard)
- Statistics
- Chatbot
- Precautions
- Symptoms

#### A.1.1.2 Search and Filters

- Search Bar to search for any district's risk level and predicted dengue cases.
- Month and year selector to view the predictions for the selected time period.

#### A.1.1.3 Dengue Risk Map

It shows the colored map of pakistan in which red color shows high risk, orange shows medium risk, green shows low risk and grey shows no data. User can click on any region on map to see a popup with risk level and also predicted cases .

#### A.1.1.4 Case Insight

It shows the current predicted cases of the your current location.

### A.1.2 Weather Insights

It shows live temperature, humidity and precipitation of user's current location.

#### A.1.2.1 Footer

A footer is present at the end of page for quick links of other pages and information about the DenguePredict.

### A.1.3 Chatbot Page

It contains predefined quick questions to assist users and an input bar which lets user to ask anything related to dengue and the bot will answer within 3-4 mins.

### A.1.4 Statistics Page

It has option to select between national, provincial and district level statistics. User can also select date range of which he wants to see the predictions

- User can select between national, provincial and district level statistics.
- Select Date Range (start and end date).
- Charts and Graphs are displayed according to the selected choices

## A.2 Mobile Application

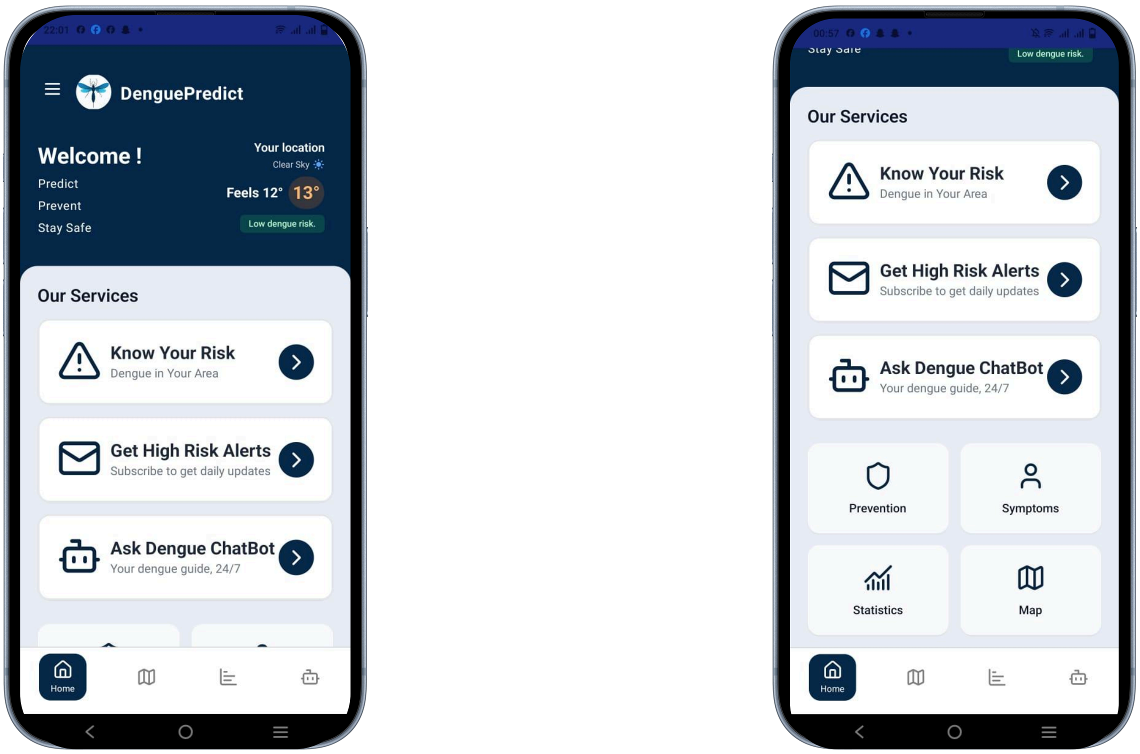


Figure A.2: App Dashboard

### A.2.1 Dashboard

It contains buttons for:

- Know Your Risk (Dengue Risk Map)
- Get High Risk Alerts
- Ask Dengue Chatbot
- Symptoms

- Precautions
- Statistics
- Drawer (Quick Links)

#### A.2.1.1 Subscribe for Alerts Page

It contains a form with input fields name and city which user needs to fill to subscribe to daily alerts if his city is at high risk.

Other app pages works similiarly to web app as described above.

## A.3 Troubleshooting

- **Map not loading:** Ensure a stable internet connection.
- **No dengue risk shown for a district:** Selected data for the chosen month or year may be unavailable; try a different time range.
- **Notifications not received:** Verify that push notifications are enabled and the device has granted notification permissions.
- **Chatbot not responding:** The external AI service may be temporarily unavailable; please retry after some time or wait for 10 mins because of overload.
- **Slow system response:** Refresh the page or restart the application to reload the latest data.

## A.4 Support Information

For assistance or queries related to the DenguePredict system, please contact: Email: denguepredict@gmail.com

# References


[1] S. Tabassum, A. Naeem, A. Nazir, F. Naeem, S. Gill, and S. Tabassum. Year-round dengue fever in pakistan, highlighting the surge amidst ongoing flood havoc and the covid-19 pandemic: a comprehensive review. *Annals of Medicine and Surgery (Lond)*, 85(4):908–912, April 2023. Cited on p. 1.

[2] U. Khan and S. Azeem. The rising toll of dengue cases in pakistan every year: An incipient crisis. *Annals of Medicine and Surgery (Lond)*, 76:103549, March 2022. Cited on p. 2.

[3] Xinyi Lu, Su Yean Teh, Chai Jian Tay, Nur Faeza Abu Kassim, Pei Shan Fam, and Edy Soewono. Application of multiple linear regression model and long short-term memory with compartmental model to forecast dengue cases in selangor, malaysia based on climate variables. *Infectious Disease Modelling*, 10(1):240–256, 2025. Cited on p. 2.

[4] W. A. M. Ali, J. M. A. S. Obaid, A. Alsolihy, N. M. K. Almekhlafy, W. S. Alqadasy, and L. N. Alabsi. Dengue fever knowledge and awareness among university students in taiz governorate, yemen: A cross-sectional study. *Health Science Reports*, 8(7):e71112, July 2025. Cited on p. 2.

[5] José G Rigau-Pérez, Gary G Clark, Duane J Gubler, Paul Reiter, Eduard J Sanders, and A Vance Vorndam. Dengue and dengue haemorrhagic fever. *The lancet*, 352(9132):971–977, 1998. Cited on p. 5.

[6] Zhilin Zeng, Juan Zhan, Liyuan Chen, Huilong Chen, and Sheng Cheng. Global, regional, and national dengue burden from 1990 to 2017: A systematic analysis based on the global burden of disease study 2017. *EClinicalMedicine*, 32, 2021. Cited on p. 5.

[7] Xiang Chen and Paula Moraga. Assessing dengue forecasting methods: a comparative study of statistical models and machine learning techniques in rio de janeiro, brazil. *Tropical Medicine and Health*, 53(1):52, 2025. Cited on p. 6.

[8] Corey M Benedum, Osama ME Seidahmed, Elfatih AB Eltahir, and Natasha Markuzon. Statistical modeling of the effect of rainfall flushing on dengue transmission in singapore. *PLoS neglected tropical diseases*, 12(12):e0006935, 2018. Cited on p. 6.

[9] Hani M Aburas, B Gultekin Cetiner, and Murat Sari. Dengue confirmed-cases prediction: A neural network model. *Expert Systems with Applications*, 37(6):4256–4260, 2010. Cited on p. 6.

[10] Ejaz Mahmood Ahmad Qureshi, Amtul Bari Tabinda, and Seemal Vehra. Predicting dengue outbreak in the metropolitan city lahore, pakistan, using dengue vector indices and selected climatological variables as predictors. *J Pak Med Assoc*, 67(3):416–421, 2017. Cited on pp. 6, 7, and 8.

[11] Michael A Johansson, Nicholas G Reich, Aditi Hota, John S Brownstein, and Mauricio Santillana. Evaluating the performance of infectious disease forecasts: A comparison of climate-driven and seasonal dengue forecasts for mexico. *Scientific reports*, 6(1):33707, 2016. Cited on pp. 7 and 9.

[12] Antoine Adde, Pascal Roucou, Morgan Mangeas, Vanessa Ardillon, Jean-Claude Desenclos, Dominique Rousset, Romain Girod, Sébastien Briolant, Philippe Quenel, and Claude Flamand. Predicting dengue fever outbreaks in french guiana using climate indicators. *PLoS neglected tropical diseases*, 10(4):e0004681, 2016. Cited on pp. 7 and 9.

[13] Magali Teurlai, Christophe Eugene Menkès, Virgil Cavarero, Nicolas Degallier, Elodie Descloux, Jean-Paul Grangeon, Laurent Guillaumot, Therese Libourel, Paulo Sergio Lucio, Françoise Mathieu-Daudé, et al. Socio-economic and climate factors associated with dengue fever spatial heterogeneity: a worked example in new caledonia. *PLoS neglected tropical diseases*, 9(12):e0004211, 2015. Cited on p. 7.

[14] Feng-Sheng Chang, Yao-Ting Tseng, Pei-Shan Hsu, Chien-Da Chen, I-en B Lian, and Deng-Yuan Chao. Re-assess vector indices threshold as an early warning tool for predicting dengue epidemic in a dengue non-endemic country. *PLoS Neglected Tropical Diseases*, 9(9):e0004043, Sep 2015. Cited on p. 7.

[15] Pei-Ju Tsai and Hsu-Jen Teng. Role of *aedes aegypti* (linnaeus) and *aedes albopictus* (skuse) in local dengue epidemics in taiwan. *BMC Infectious Diseases*, 16(1):662, Nov 2016. Cited on p. 7.

[16] Junaid Khan, Imran Khan, and Ijaz Amin. A comprehensive entomological, serological and molecular study of 2013 dengue outbreak of swat, khyber pakhtunkhwa, pakistan. *PLoS ONE*, 11(2):e0147416, Feb 2016. Cited on p. 7.

[17] Shuqing Sang, Wenwu Yin, Peng Bi, Hong Zhang, Chun Wang, Xiao Liu, Bing Chen, Wenwu Yang, and Qiyong Liu. Predicting local dengue transmission in guangzhou, china, through the influence of imported cases, mosquito density and climate variability. *PLoS ONE*, 9(7):e102755, Jul 2014. Cited on p. 8.

[18] Yien Ling Hii, Hua Zhu, Norman Ng, Lee Ching Ng, and Joacim Rocklöv. Forecast of dengue incidence using temperature and rainfall. *PLoS Neglected Tropical Diseases*, 6(11):e1908, 2012. Cited on p. 9.

[19] Matthew D. Eastin, Eric Delmelle, Idilio Casas, Joshua Wexler, and Charles Self. Intra- and interseasonal autoregressive prediction of dengue outbreaks using local weather and regional climate for a tropical environment in colombia. *American Journal of Tropical Medicine and Hygiene*, 91(3):598–610, September 2014. Cited on p. 9.

[20] Van-Hung Nguyen, Tran Thi Tuyet-Hanh, James Mulhall, Hoang Van Minh, Tuan Quang Duong, Nguyen Van Chien, et al. Deep learning models for forecasting dengue fever based on climate data in vietnam. *PLoS Neglected Tropical Diseases*, 16(6):e0010509, 2022. Cited on p. 9.

[21] Kathleen M. O'Reilly, Emilie Hendrickx, Dinar D. Kharisma, Nandyan N. Wilastonegoro, Lauren B. Carrington, Iqbal R. F. Elyazar, Adam J. Kucharski, Rachel Lowe, Stefan Flasche, David M. Pigott, Robert C. Reiner, W. John Edmunds, Simon I. Hay, Laith Yakob, Donald S. Shepard, and Oliver J. Brady. Estimating the burden of dengue and the impact of release of wmel wolbachia-infected mosquitoes in indonesia: a modelling study. *BMC Medicine*, 17(1):172, 2019. Cited on p. 44.