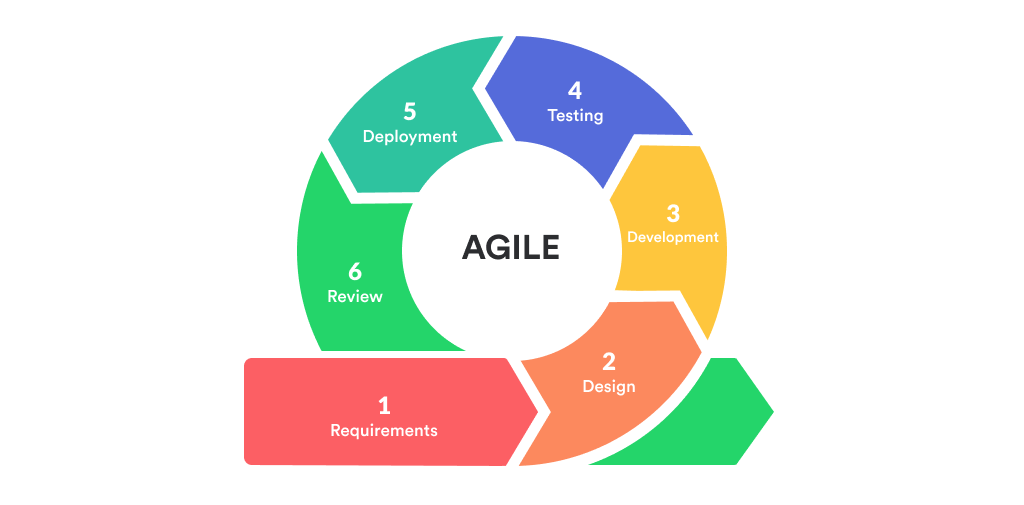
**CHAPTER 3 – METHODOLOGY**

|  |  |  |
| --- | --- | --- |
| **3.1** | Agile Software Development | |
| **3.2** | Project Framework | |
| **3.3** | **Phase 1: Requirement Analysis** | |
|  | **3.3.1** | Business Requirements |
|  | **3.3.2** | Data Collection |
|  | **3.3.3** | Data Source |
|  | **3.3.4** | Data Preprocessing |
| **3.4** | **Phase 2: Design** | |
|  | **3.4.1** | System Architecture |
|  | **3.4.2** | Use Case Diagram |
|  | **3.4.3** | Flowchart Diagram |
|  | **3.4.4** | Prototype User Interface   1. Landing Page 2. Hourly Weather Page 3. Daily Weather Page 4. Monthly Weather Page 5. Activities Recommendation Page |
| **3.5** | **Phase 3: Development** | |
|  | **3.5.1** | Hardware Requirement |
|  | **3.5.2** | Software Requirement |
|  | **3.5.3** | Model Development |
|  | **3.5.4** | Prototype Development |
| **3.6** | **Phase 4: Testing** | |
|  | **3.6.1** | Functionality |
|  | **3.6.2** | Usability |
| **3.7** | **Gantt Chart and Milestones** | |
| **3.8** | **Summary** | |

**RESEARCH METHODOLOGY**

**3.1 Agile Software Development**

The gradual and iterative approach to software development is known as the Agile development methodology. It begins with requirements, which are collected and assessed to ensure understanding of the project's nature. The design phase releases system architecture and detailed designs, allowing for flexibility and modularity. The development phase involves coding and feature implementation in incremental builds, allowing for continuous progress tracking. The testing phase involves rigorous testing to identify and resolve bugs, ensuring the system meets specified requirements and meets high quality standards. These phases allow continuous repetition, feedback, and improvement. However, as an undergraduate student running this project, the other two phases are not required for undergraduate students because developing and maintaining this project takes a long time. In Figure 3.1, the project model is presented.



**Figure 3.1** Agile Model

(Source: https://serenagray2451.medium.com/what-is-the-agile-methodology-in-software-development-c93023a7eb85)

**3.2 Project Framework**

A diagram of a process

Description automatically generated

**Figure 3.2** Overall project framework in Agile Methodology

**3.3 Phase 1: Requirement Analysis**

### **3.3.1 Business Requirements**

The phase focuses on defining the goals and expectations for a weather prediction system. It begins by identifying key problem statements and defining the project scope. This ensures everyone understands the system's limitations and aims. This phase also analyses previous studies and research to learn from successes and mistakes. This subtopic establishes the foundation for the project by establishing clear objectives and leveraging existing knowledge.

### **3.3.2 Data Collection**

To train and evaluate machine learning models for a weather prediction system, the data collecting step entails obtaining historical and current weather data from multiple sources. This extensive data set encompasses a wide range of weather patterns and situations, offering valuable insights into extended trends and guaranteeing precise forecasts. The accuracy and dependability of the model in predicting upcoming weather occurrences can only be increased with a vast and diverse dataset.

### **3.3.3 Data Source**

Finding reliable sources of meteorological data is essential to ensure the accuracy of a weather prediction system. For consistency and accuracy to be guaranteed, data integrity is essential. The system can produce precise and dependable weather forecasts by choosing trustworthy sources. For this project, the source of data will be from Visual Crossing or Kaggle website.

### **3.3.4 Data Preprocessing**

In order for the machine learning models to properly extract knowledge from the collected data, it must first be cleaned up and organized. This covers necessary tasks including normalizing the data, fixing missing data, and eliminating abnormalities. The obtained meteorological data is refined and arranged in an analysis-ready format during preprocessing. The primary step in this phase is the shrinking of the data, which maintains its quality and reduces the predicted mistakes caused by the machine learning models.

**3.4 Phase 2: Design**

The application's design takes importance at this stage. This section will cover the four main design categories for this application: system architecture, the use case diagram, the flow diagram with a flowchart, and the UI design. Once the design phase is complete, the application will be prepared for development.

**3.4.1 System Architecture**

A diagram of a data flow

Description automatically generated

**Figure 3.3** Overview of System Architecture

Based on Figure 3.3, it shows the system architecture of this project. It involves a user interface, API interaction, database storage, prediction and recommendation, and result retrieval. The user enters their location, selects the desired prediction type, and the system sends location data to the API for retrieval. The received weather details are stored in a database, and the recommendation engine calculates and provides weather-related recommendations, ensuring accurate and timely weather prediction and activity suggestions.

**3.4.2 Use Case Diagram**

A use case diagram is used to display the overall design of this application. It is used to determine the users along with the activities that the application itself is capable of. The following figure illustrates the overview of the application.

A diagram of weather forecasting

Description automatically generated

**Figure 3.4** Application Design Overview

**3.4.3 Flowchart Diagram**

The system's process will be described in this section. A flowchart will be used to illustrate the system's workflow by describing the steps involved.

A diagram of a weather forecast

Description automatically generated

**Figure 3.5** The flow of the user throughout usage of the application.

**3.4.4 User Interface (UI) Design**

For the UI design, the application will consist of 6 main pages. The following shows the respective pages of the landing page.

1. Landing Page

A screenshot of a computer

Description automatically generated

**Figure 3.6** Overview of User Landing Page

The website redirects users to the weather prediction system landing page, featuring a clean design with a top navigation bar, location input, and tabs for daily, weekly, monthly, and activities. A prominently displayed map shows the selected location. The main section provides current weather conditions in Shah Alam, Malaysia, while the footer includes links to company information, products, and services.

1. Hourly Weather Page

A screenshot of a weather forecast

Description automatically generated

**Figure 3.7** Overview of Hourly Weather Page

The hourly weather page provides detailed weather information for each hour. For example, in the above figure, it displays the weather for this present hour: temperature, what the temperature feels like, wind, UV index, cloud, humidity and rainfall. This is followed by an hour-by-hour prediction for the upcoming few hours.

1. Daily Weather Page

A screenshot of a weather forecast

Description automatically generated

**Figure 3.8** Overview of Daily Weather Page – Night

A screenshot of a weather report

Description automatically generated

**Figure 3.9** Overview of Daily Weather Page – Day

Users will be provided with daily information like wind, percentage of humidity, cloud cover, and any probability of precipitation or thunderstorms.

1. Monthly Weather Page

A screenshot of a calendar

Description automatically generated

**Figure 3.10** Overview of Monthly Weather Page

The monthly weather page displays a calendar view of the weather prediction for selected month. Each day shows the expected high and low temperatures along with a weather icon indicating the general conditions (e.g., sunny, partly cloudy, rain). The layout is structured with days of the week labelled at the top and dates filling the grid below, providing an at-a-glance view of the month's weather patterns. Navigation options at the top allow users to switch between different time frames (Today, Hourly, Daily, Monthly) and access other features like Activities.

1. Activities Recommendation

A screenshot of a computer

Description automatically generated

**Figure 3.11** Overview of Activities Recommendation Page

Figure 3.11 presents an activity recommendation page tailored for farmers, indicating which activities are advisable and which ones should be avoided on a given day.

**3.5** **Phase 3: Development**

The project is being developed using hardware and software, with minimal requirements met to ensure smooth development and avoid installation issues. The development phase is crucial for achieving the project's third purpose, involving creating a prototype based on the design, which is the final delivery.

**3.5.1 Hardware Requirement**

To develop the project, the minimum hardware requirements needed are as follows:

* Intel(R) Core 2 Duo CPU @ 2.00 GHz, 2.00 GB of RAM
* Any good quality personal computer such as Dell, Lenovo, Asus, Acer, HP or Compaq
* 320 GB Hard Disk Driver
* Keyboard and mouse

**3.5.2 Software Requirement**

* Windows 7 or newer
* BlueJ, NetBeans, or Notepad++ for system development coding
* Microsoft Office Word
* Google chrome
* Firebase
* Canva

**3.5.3 Model Development**

To achieve optimal performance, key procedures must be followed when developing a machine learning model. Selecting a suitable algorithm is the first stage; the Random Forest method, which manages intricate, non-linear relationships in data, is one such way. The model is trained using a large amount of historical meteorological data, which is analysed for patterns and connections. Using new datasets and hyperparameter tweaking techniques, performance is evaluated. The process of cross-validation is used to guarantee precision and dependability. The updated model is set up with the required data streams and linked to a real-time system in order to deliver precise and fast weather forecasts. The model is anticipated to become the most accurate and dependable for real-time weather prediction due to this recursive process.

**3.5.4 Prototype Development**

Prototype development is a crucial stage in the development of a system, where essential hardware and software components are assembled to create a functional model. This stage allows engineers to test and refine the design, ensuring it meets project requirements and performs as expected. Prototype development involves iterations to experiment with different configurations, interfaces, and functionalities, providing stakeholders with a hands-on understanding of the system's capabilities. By simulating real-world conditions, prototype development helps identify and resolve issues early in the development process.

**3.6 Phase 4: Testing**

Functionality testing is used to cross-check whether an individual feature of a system complies with requirements and operates as expected. It involves going through each facet of the program step by step, from backend activities to frontend operations that confirm its capability. Test cases are conducted to ensure the system's functionality, including recommendation creation, forecast accuracy, and data intake. This ensures robustness and reliability, ensuring customer satisfaction and accurate weather forecasts. Thorough testing helps identify and resolve issues, ensuring system functionality as intended. The test case of this project is shown in Table 3.1 below.

**Table 3.1** Test Cases

|  |  |  |
| --- | --- | --- |
| **Component** | **Test Case** | **Expected Outcome** |
| View Real-Time Weather | Input correct location. | Display accurate real-time weather for the specified location. |
| Refresh page. | Real-time weather updates correctly with current data. |
| Access Weather Prediction | Request hourly prediction. | Provide accurate and detailed hourly weather prediction for the location. |
| Request daily prediction. | Display accurate and detailed daily weather prediction for the location. |
| Request monthly prediction. | Shows accurate and detailed monthly weather prediction for the location. |
| Receive Personalize Recommendations | Check that users receive tailored recommendations based on their preferences and current weather conditions. | Users receive relevant and personalized recommendations consistently. |

**3.7 Gantt Chart and Milestones**

The project consists of several phases that are expected to be completed within a timeframe. Figure below shows the scheduled phase.

**Table 3.1** Gantt Chart for The Project Proposal Semester 5

**A screenshot of a computer

Description automatically generated**

**Table 3.2** Gantt Chart for The Project Proposal Semester 6A screenshot of a computer

Description automatically generated

**3.8 Summary**

The chapter introduces a step-by-step procedure to develop a weather forecasting system through agile software development. At the beginning, a need-based requirement is stressed; it includes the necessity of defining business requirements, using various data sources, converting the data through preprocessing, and unity with the system's architecture. The user interface design is the key feature, and it is furnished by key pages like the landing page and the one showing the weather of the current hour, the current day, and the next month. The design includes hardware and software parts, model development using machine learning algorithms, and activity recommendation features. The research paper mentions the key importance of the research, the outcome of the system in terms of the user interface, and the use of a Gantt chart in the project planning phase. To sum up, the approach shows a clear-cut technique for creating a trustworthy and easily accessible weather prediction system with very accurate predictions.