



Weather Forecast WordPress Theme

- Easy to Customize
- Clean & Modern Design
- 24/07 Support Team

15+
Demo



WEATHER FORECAST

TEAM MEMBERS :

Mena Mohamed Osman 221100521
Shrouq Hisham Salman 223108628
Hajar Mohammed 222200040
Asser Mohamed Ali 222102487
Jomana Sami Abdelaziz 222100458

DR: Mohamed Abdallah
CSE251: Software Engineering

TABLE OF CONTENTS

1. Abstract.....	1
2. Introduction	2
2.1 Purpose... ..	2
2.2 Scope	3
2.3 Overview	3
3. SOFTWARE PROCESS MODEL	3
3.1 SDLC Overview	3
3.2 Software Process Model.....	3
4. General section.....	3
4.1 Problem statement.....	3
4.2 Drawbacks of existing System.....	3
4.3 Proposed system	3
5. Functional Requirements	4
5.1 Weather data requirements	4
5.2 User log in	4
5.3 Searching... ..	4
6. Interface Requirements.	4

6.1	<u>GUI.....</u>	<u>4</u>
6.2	<u>Software Interface.....</u>	<u>4</u>
6.3	<u>Communication interface... ..</u>	<u>4</u>
6.4	<u>Performance Requirements</u>	<u>5</u>
6.5	<u>Design Constraints.....</u>	<u>5</u>
6.6	<u>Non-functional Requirements.....</u>	<u>5</u>
6.7	<u>Performance Requirements</u>	<u>5</u>
	<u>6.7.1 performance requirement</u>	<u>5</u>
6.8	<u>UML Diagrams</u>	<u>10</u>
6.9	<u>Operational Scenarios</u>	<u>10</u>
6.10	<u>Appendices.....</u>	<u>10</u>

DESCRIPTIVE ABSTRACT

This report conducts , or showa a study on the guide of building a weather forecast platform , where the platform allows people to get know the weather forecast based on their location or other locations . the concern of this report is to show the process undergone so the product can be delivered to the user in the highest quality possible .

This report is going to show how the product is made , passing through every stage through the software development life cycle (SDLC) shows the main processes , and activities undertaken to make the software ; starting from collecting requirments and specifications from the client , to deliver the product , passing through many activites such as : design , coding , integration and maintenance .

Software Process Model may play a big role in delivering the product to the customer , as they are set of orders and structured actions taken by order for software development . there are several Software Process Model , each have different process flow ,

but in the end ,they are following the same activities of planning design , and deployment . in this report we will show how the softwre is being built using five software process models which are Waterfall Model, V Model , Transformation Model, Spiral Model, Evolutionary Model showing the output of each stage .

I. INTRODUCTION

~~Weather forecasting as a mobile application~~

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change. The role of Technology has been remarkable in the field of weather forecasting. Weather data is not only necessary for researchers or scientists, ordinary people can be benefitted from it as well. People nowadays are feeling the necessity of weather data as well. There are a variety of weather mobile apps in Google Play and the App store. Those apps have great features and functionalities to satisfy users. However, only a few of them have friendly user interface and human centered interactions, which means that a lot of them might be difficult to be navigated even though they provide enough functionalities. It is not convenient for new users. Therefore, we would like to do improvements on weather mobile apps. It is basically for Apple smart phones and tablets

1.1 OVERVIEW

The project, the “WEATHER FORECASTING APP” is a cutting edge and versatile Weather application fundamentally designed to help people to travel long distances connect with their near place ones in a very friendly, easy and hassle-free manner. This application is compatible with all the Android versions. And with said that, anyone owing their own android phone, the Weather Report application can create a huge difference in hooking them up with others provided, they also have the same application loaded on their device or PC. It is very simple to use and easy to install that can be downloaded directly. When it comes to the features, they all are extremely engaging and they keep updating periodically back to back. The extent of fun in using this Weather application is beyond comparison. This application is highly accommodative for quick technological updates and integration. The application will have the features like providing Forecast, Humidity, Pressure etc. It also provides Forecast Weather for 7 Days all over the world. In today’s busy routine life, the users of this Weather Report application will be just a click away to access all the basic requirements in today’s day to day routine of the humans. The users will have access to the latest Weather updates i.e. the burning Weather in all around the world. Secondly, the users will have the access to the weather forecast before going out of their home with the support of the GPS in their smart-phones. Compared to the other applications this Weather Report application will prove to be an amazing user friendly application with lots of more features.

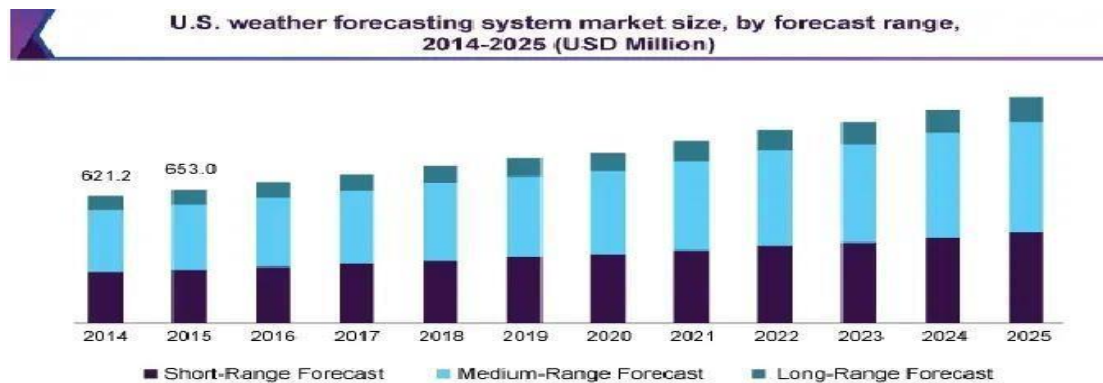
1.2 PROJECT AIM

To make areal time weather application that takes user’s exact location and provides weather forecast for the day and upcoming days also. We also tried to design a simple but visual UI that provides comprehensive data. Also, the application provides

suggestions to users based on weather conditions. And lastly, user can search and access data for custom locations (string based)

1.3 SCOPE

The global weather forecasting system and solutions market size was valued at USD 2.51 billion in 2016. It is expected to post a CAGR of 7.1% over the forecast period. These systems help enterprises in gaining real-time insights into atmospheric conditions, which in turn, enables enterprises to carefully plan all weather-sensitive operations to ensure security, sustainability, safety, and cost efficiency. Factors such as growing sea and air transportation, increasing stringency of norms pertaining to environmental protection, and high dependency on rainfall for water supply are among the key trends stimulating market growth.



II. SOFTWARE PROCESS MODEL

A. SDLC Overview

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.

The software development lifecycle (SDLC) outlines several tasks required to build a software application. The development process goes through several stages as developers add new features and fix bugs in the software.

The details of the SDLC process vary for different teams. However, we outline some common SDLC phases below.

1) Plan

The planning phase typically includes tasks like cost-benefit analysis, scheduling, resource estimation, and allocation. The development team collects requirements from several stakeholders such as customers, internal and external experts, and managers to create a software requirement specification document.

The document sets expectations and defines common goals that aid in project planning. The team estimates costs, creates a schedule, and has a detailed plan to achieve their goals.

2) Design

In the design phase, software engineers analyze requirements and identify the best solutions to create the software. For example, they may consider integrating pre-existing modules, make technology choices, and identify development tools. They will look at how to best integrate the new software into any existing IT infrastructure the organization may have.

Implement

In the implementation phase, the development team codes the product. They analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

3) Test

The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

4) Deploy

When teams develop software, they code and test on a different copy of the software than the one that the users have access to. The software that customers use is called *production*, while other copies are said to be in the *build environment*, or testing environment.

Having separate build and production environments ensures that customers can continue to use the software even while it is being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environment configuration, and installation.

5) Maintain

In the maintenance phase, among other tasks, the team fixes bugs, resolves customer issues, and manages software changes. In addition, the team monitors overall system performance, security, and user experience to identify new ways to improve the existing software.

B. Software Process Model

A software development lifecycle (SDLC) model conceptually presents SDLC in an organized fashion to help organizations implement it. Different models arrange the SDLC phases in varying chronological order to optimize the development cycle.

1) Waterfall

The waterfall model arranges all the phases sequentially so that each new phase depends on the outcome of the previous phase.

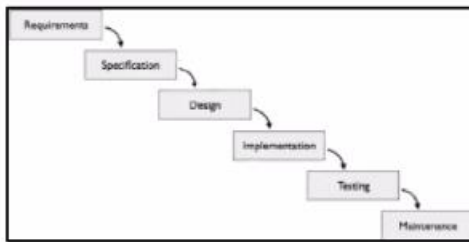


Fig. 1. shows the illustrated representation of the Waterfall model, as it shows the linear flow of stages, which is the same as a real waterfall.

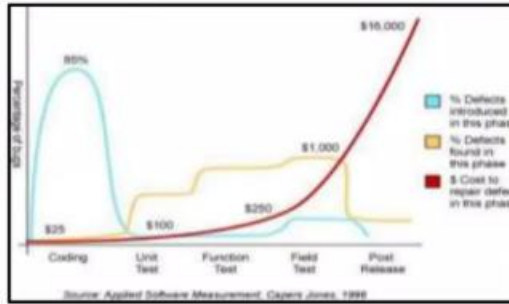


Fig. 2. shows the cost defects expected that can be by making errors in Waterfall stages.

Table 1 ADVANTAGE AND DISADVANTAGES OF WATERFALL MODEL

Advantages	Disadvantages
Easy to apply and explained	Product delivery is sentenced to single stricted deadline
Well and simple structure of phases	Costly and requires a lot of time
Each phase has specific deliverables	Forces planning after limited analysis
Works for big projects where requirements are well understood from the start	Hard to apply changes after moving to a next stage
Processes and results are documented at every stage end	No continous customer involvement in reviewing output of each phase

2) Evolutionary Model

This model has several models which inherited the main concept from him, but they had several differences. The main concept implies that the proposed system is broken down to small blocks and working on it through incremental iterations in order to expand the system, that is why it is called evolutionary where the system undergoes evolution process.

a) The Prototype Evolutionary Model

Prototype Evolutionary Model proposes to create a small block of the system by gathering requirements for this part, then create a prototype and show it on the customer, then recording the customer's feedback on this prototype, then make improvements to this prototype if required, and finally push the created incremented product to the system.

b) The Agile Model

Agile Software Process Model came as a rescue to Software Crisis as several problems faced software engineers and customers when making a software

1. Customer does not check or see the proposed product before the testing phase in Waterfall model
2. You could be in deployment phase and realize that product has no viable value due to change in market's perspectives or organizational directions
3. The product had architectural flow that prevented it from being deployed
4. It uses top down management which can lead to overbudget

Agile rescue and proposed solutions

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documents
3. Customer collaboration over contract negotiation
4. Responding to change over following plan

Agile model consists of several frameworks like Scrum that helps build complex products. Scrum builds products incrementally and iteratively in time box periods of 30 calendar days. It consists of Sprints; Sprints comprises one product iteration. Sprint begins with Sprint Planning where product owner works with development team to mark which product backlog will be built.

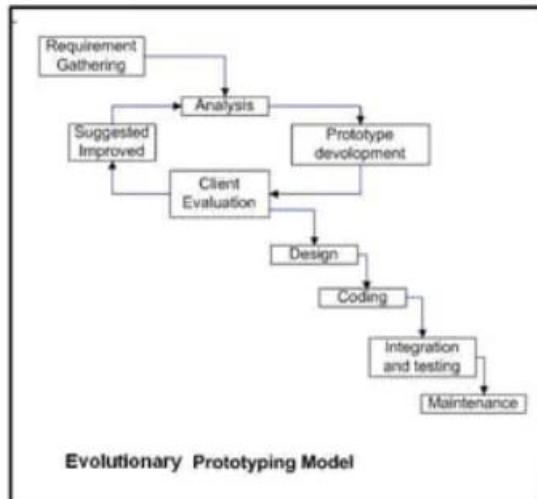


Fig. 2. Shows illustrated representation of Prototyping Evolutionary Model

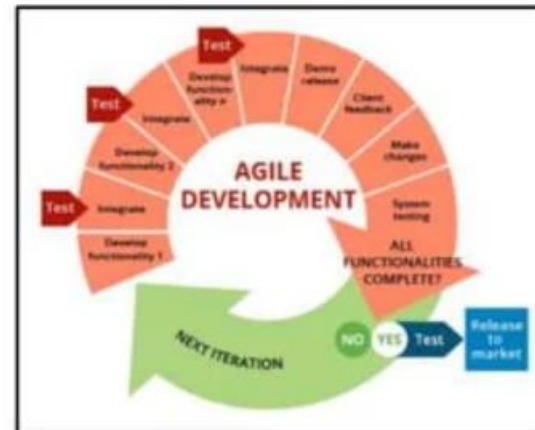


Fig. 4. Shows illustrated representation of Agile Model

TABLE II. ADVANTAGES VS DISADVANTAGES PROTOTYPING EVOLUTIONARY MODEL

Advantages	Disadvantages
Reduce cost and time	User confusion of prototype and final release of the system
Improved user and product owner involvement	Developer can misunderstand of user objectives
Users give corrective feedback	Excessive development time of prototypes
A more accurate final product	Expenses of developing a lot of prototypes might be costly

TABLE III. ADVANTAGES VS DISADVANTAGES AGILE MODEL

Advantages	Disadvantages
Decrease time required to present some system features	Scalability
Continuous interviews and plannings with users so no guesswork	Easy to understand, hard to master
High quality product in least time duration	Documentation done at later stages
Keeps everything simple	Not suggested for large scale products

3) Spiral Model

This model inherits the concept of the Evolutionary Model, in addition to being a combination of Prototyping and Waterfall models, but what makes it more different is depending more on Risk Analysis strategy. Its focus is to identify and eliminate high risk problems faced that might appear suddenly. The spiral model undergoes 4 stages.

1. Determine main objectives and design alternatives
2. Evaluate alternatives, identify, and resolve risks
3. Develop and verification of next level product
4. Plan for next phases

A Spiral model is divided into four frameworks that were mentioned above, each of these frameworks represents a segment of the spiral path. As the spiral process begins, the software engineering team members start performing actions that are implied by the circle around the spiral in a clockwise direction. First cycle starts with gathering requirements and specifications from customers and objectives are identified and analyzed, then alternative solutions are proposed for this phase. In the second cycle, the team analyzes each solution then pick the best one, then identify risks associated with that solution and try to resolve it with the best strategies, then deliver a prototype for the solution presented before. During the third cycle, the prototype is undergone through development and verification by testing phase, then deliver the next version of that system. In the fourth cycle, customers evaluate and present feedback on the version presented, then planning for next phase start.

TABLE IV. ADVANTAGES VS DISADVANTAGES SPIRAL MODEL

Advantages	Disadvantages
Estimates because of Risk Analysis become more realistic	Costly and takes a lot of time to deliver final product
Choice of changing requirements is available	Needs skilled team to study risks
Customers have ability to view versions of the system	Time spent on planning for the versions, studying the risks, and presenting solutions is huge.
A more accurate end product.	



Fig. 4. Shows illustration of Spiral Model

4) Transformation Model

This model is concerned with application of a mathematical technique to design and implement the software. It is mostly reliable on for developing complex big systems and support the development phase of the program. It uses a strategy known as Optimization where requirements are broken down to more simple requirements so it can be easy to implement and provide a solution for eliminating problems which could be hard to overcome in other software process models.

Transformation model comprises formal specification by usage of mathematics to determine desired functionalities of the system. Transformation specifications are like Waterfall specifications, but they are converted into a mathematical module and based on functions which were defined using mathematical notations.

Steps conducted in this model

1. Requirements studied and converted into formal requirements
2. Formal specifications transformed into detailed and less abstract description
3. Repetition for step 2 until description can be executed by processor
4. Increase efficiency by more transformations

Transformations may be performed by Software Engineers or automated systems.

More actions taken in this model like recording of development history where it is more like a cost analysis stage to determine if some development methods are working properly or not. Reusable components where software engineers might use a lot of data and components in order to get to reach to results, they do not know.

The Transformation model have two approaches, property-based and model-based. The property-based specification describes every operation performed on the system and describes all the relationships, if exists among these operations. Property-based specification consists of two parts: signatures, which determines the syntax of operations done on the system and equations, which describes semantics of operations through set of equations called axioms. The model-based develops an abstract model of the system at high level and ideality, and specifies the operations performed on the abstract model.

TABLE V. ADVANTAGES VS DISADVANTAGES TRANSFORMATION MODEL.

Advantages	Disadvantages
Discovers ambiguity and inconsistency in the software requirements	Time consuming and automated systems that convert specifications are expensive
Incrementally grows in effective solutions after each iterations	Only few developers know about this model and how it works
System proven that it will work properly	Difficult to use this model for non technical personas
	Very complex

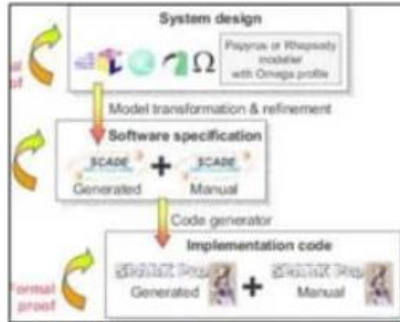


Fig. 6. Shows illustrated representation of Transformation model

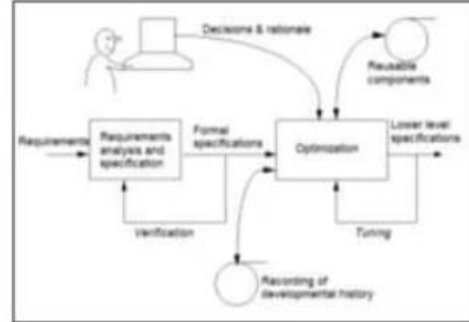


Fig. 7. Shows steps conducted in Transformation model

5) V-Shape Model

It is a model where processes are being executed in a linear flow like Waterfall model, but in addition to V-Shape manner. It is called as Verification Validation model, as it is built mostly on association of a testing phase for each development stage in the cycle, which means that each development phase has a testing phase in return. It has the same Waterfall concept where a phase starts after completing the corresponding test phase.

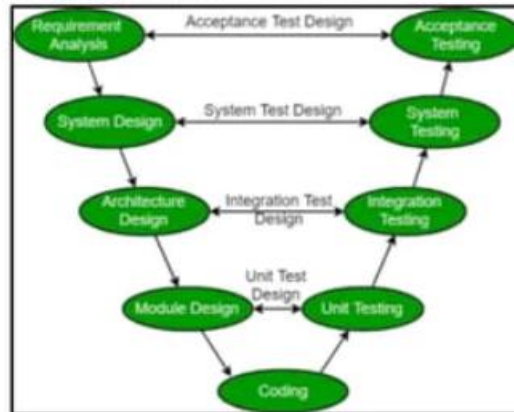


Fig. 8. Shows representation of V-Shaped Model

- **Unit Testing:** A phase executed to eliminate any errors or bugs at code.
- **Integration Testing:** Modules are integrated together, and system is going through testing to verify communication of modules among themselves.
- **System Testing:** The whole system is tested with focus if the system satisfies the requirements proposed.
- **User Acceptance Testing:** Customers test the system in user environment to check if the system is ready to be delivered to real users.

TABLE VI. ADVANTAGES VS DISADVANTAGES V-SHAPE MODEL.

Advantages	Disadvantages
Verification of each phase early before submitting the whole system	Costly and requires a lot of time
Simple and easy to understand	Does not support iterations
Project management gets to track progress accurately	Not suitable for large projects
Works well for easily understood requirements	Does not easily handle dynamic changes in requirements

6) Comparison between all models

TABLE VII. COMPARISON BETWEEN ALL MODELS

Factors	Waterfall	Evolutionary		Spiral	Transformation	V-Shape
		<i>Prototype</i>	<i>Agile</i>			
Unclear User Requirements	Yes	No	No	No	No	Yes
Complex System	No	No	Yes	No	Yes	No
Reliable System	Yes	Yes	No	Yes	Yes	No
Short Time Schedule	No	No	Yes	Yes	No	No
Strong Project Management	Yes	Yes	Yes	Yes	Yes	Yes
Costly	Yes	Yes	No	No	Yes	Yes
Skills Required	No	Yes	Yes	Yes	Yes	No
Documentations	Yes	Yes	No	Yes	No	Yes
Quality						
Popularity	Yes	No	Yes	Yes	No	No
System Performance	Yes	No	Yes	Yes	Yes	No

4.1 Problem statement

If it is true, as every scientist believes, that subsequent atmospheric states develop from the preceding ones according to physical law, then it is apparent that the necessary and sufficient conditions for the rational solution of forecasting problems are the following:

- A sufficiently accurate knowledge of the state of the atmosphere at the initial time.
- A sufficiently accurate knowledge of the laws according to which one state of the atmosphere develops from another.

4.2 Drawbacks of existing System

Weather forecasting

Advantages	Disadvantages
<ul style="list-style-type: none">• Farmers can know when to plant or harvest their crops	<ul style="list-style-type: none">• Weather is extremely difficult to forecast correctly
<ul style="list-style-type: none">• People can choose where and when to take their holidays to take advantages of good weather	<ul style="list-style-type: none">• It is expensive to monitor-so many variables from so many sources
<ul style="list-style-type: none">• Surfers known when large waves are expected	<ul style="list-style-type: none">• The computers needed to perform the millions of calculations necessary are expensive
<ul style="list-style-type: none">• Regions can be evacuated if hurricanes or floods are expected	<ul style="list-style-type: none">• The weather forecasters get blamed if the weather is different from the forecast

Proposed system

4.2.1 User registration: Administrator permit user for registration; companies can also do registration.

4.2.2 Inserting the details: user and companies have to insert their details.

4.2.3 Updating the details: As time user and companies have to update their details.

4.2.4 Event Details: Administrator update the weather.

4.2.5 Recruitment Process: Every 3-hour updating weather.

5 Functional Requirements

5.1 Weather Data Requirements

Current Weather Data:

The system shall provide current weather information based on the user's location.

The system shall allow users to input a city name to get current weather data for that location.

The system shall display weather parameters such as temperature, humidity, wind speed, and weather conditions (e.g., sunny, cloudy, rain).

5.2 Additional Features

Weather Alerts:

The system shall display weather alerts and warnings provided by the OpenWeather API.

The system shall allow users to configure notifications for severe weather alerts.

Historical Weather Data:

The system shall allow users to view historical weather data for a specified location and date range.

5.3 Performance Requirements

API Integration:

The system shall fetch weather data from the OpenWeather API in real-time.

The system shall handle API rate limits and errors gracefully.

Performance and Scalability:

The system shall respond to user queries within a few seconds.

The system shall be able to handle multiple simultaneous users without significant performance degradation.

5.4 API Key Management:

The system shall securely manage the OpenWeather API key and ensure it is not exposed to unauthorized users.

6.1 Searching

- Location

6.2 Graphical User Interface (GUI)

There will appear a container search box where user should enter his/her location then it will display the weather details like temperature, humidity, and wind speed

And if the location is searching for does not exist a message will appear this message the location doesn't exist

6.3 Software Interface

- Programming Language: python, HTML, CSS
- We used the API

6.4 Communication interface

6.4.1 This website supports all types of web browsers.

6.4.1.1 User will be informed about placement via entering locations name then it will access open weather through API key .

7.1 Performance requirements

Response Time:

Response time will be less than 1s.

7.2 Reliability Requirements

1. Availability:

The system shall have an uptime of 99.9% to ensure users can access weather information reliably.

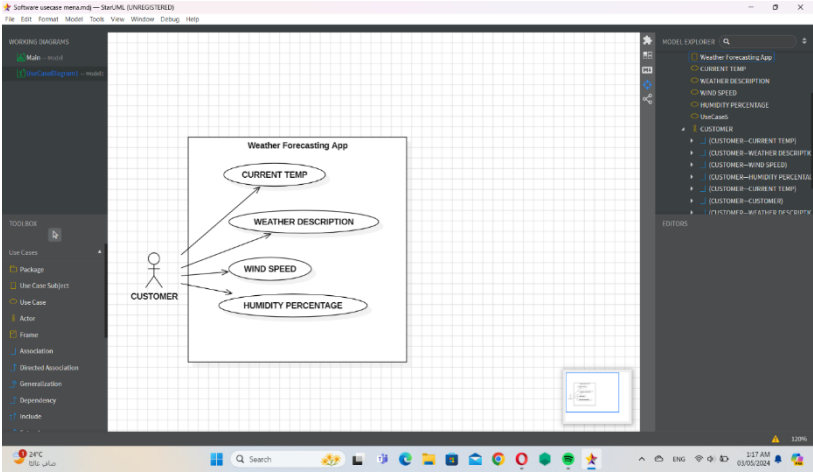
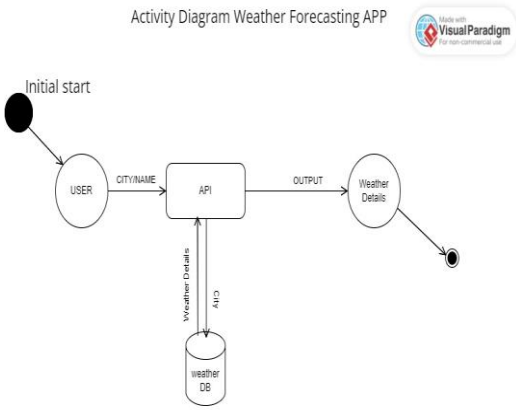
2. Data Accuracy:

The system shall ensure that the weather data presented is up-to-date and accurate as provided by the OpenWeather API.

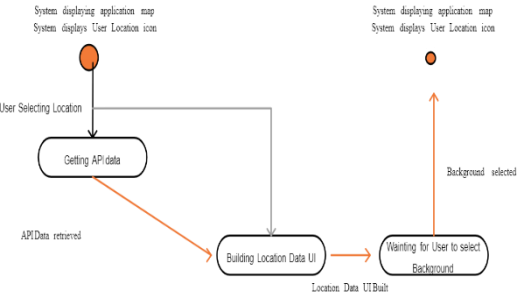
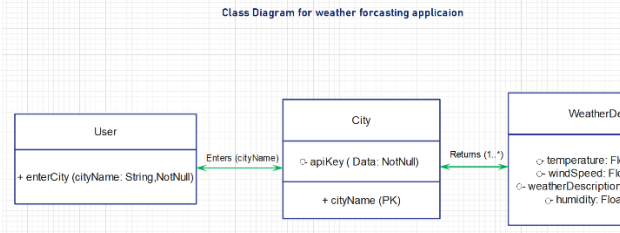
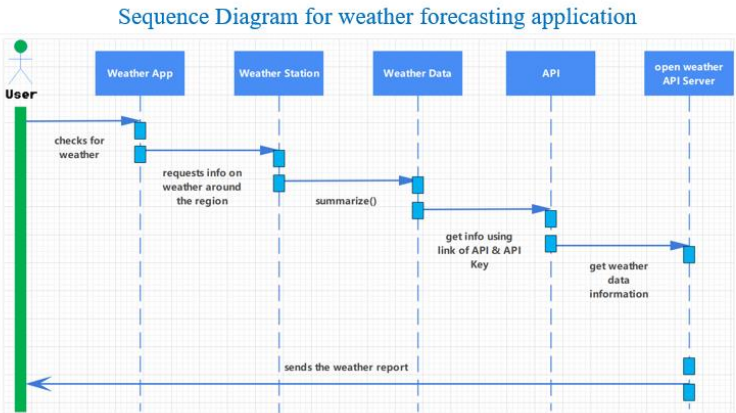
8 Design Constraints

- Software must be flexible so that user can easily update their location

9. UML diagrams

Points of comparison	Use-case diagram	Activity diagram
		 <p>Activity Diagram Weather Forecasting APP</p> <p>Initial start</p> <p>USER</p> <p>CITYNAME</p> <p>API</p> <p>OUTPUT</p> <p>Weather Details</p> <p>Weather DB</p> <p>CITY</p> <p>mena osman</p>

Summary	A use-case diagram illustrates user interactions and system functionalities	This activity diagram captures the sequence of actions and decisions involved in viewing weather data, providing a clear overview of the process flow.
---------	---	--

Points of comparison	State diagram	Class diagram
Summary	<p>state diagram depicts object states and transitions</p>  <pre> stateDiagram-v2 [*] --> S1: User Selecting Location S1: System displaying application map System displays User Location icon S1 --> S2: API Data returned S2: Building Location Data UI S2 --> S3: Location Data UI Built S3: Waiting for User to select Background S3 --> S4: Background selected S4: System displaying application map System displays User Location icon </pre>	<p>while a class diagram models the system's static structure including classes and their relationships</p>  <pre> classDiagram class User { +enterCity(cityName: String, NotNull) } class City { <> apiKey (Data: NotNull) + cityName (PK) } class WeatherData { <> temperature: Float <> windSpeed: Float <> weatherDescription <> humidity: Float } User --> City : Enters (cityName) City --> WeatherData : Returns (1..*) </pre> <p>sequence diagram shows the sequence of interactions between objects to accomplish specific tasks.</p>  <pre> sequenceDiagram actor User participant WeatherApp as Weather App participant WeatherStation as Weather Station participant WeatherData as Weather Data participant API as API participant OpenWeatherAPI as open weather API Server User->>WeatherApp: checks for weather activate WeatherApp WeatherApp->>WeatherStation: requests info on weather around the region deactivate WeatherApp activate WeatherStation WeatherStation->>WeatherData: summarize() deactivate WeatherStation activate WeatherData WeatherData->>API: get info using link of API & API Key deactivate WeatherData activate API API->>OpenWeatherAPI: get weather data information deactivate API activate OpenWeatherAPI OpenWeatherAPI-->>User: sends the weather report deactivate OpenWeatherAPI deactivate API deactivate WeatherData deactivate WeatherStation </pre>

10. Scenario: User Searches for Weather Forecast for a Specific City

User Action: The user enters the name of a city in the search bar.

System Response: The system sends a request to the OpenWeather API to retrieve weather data for the specified city.

System Response: The system receives the weather data from the OpenWeather API.

System Action: The system displays the current weather data and the 7-day weather forecast for the specified city, including daily temperature highs and lows, precipitation chances, and general conditions.

11. Appendices

This software will make easy to analyse weather. also, user show weather anytime and anywhere.

Reference:<https://www.bing.com/search?q=weather+forecasting+syst+em+prelimi>