CMSE321

Intermediate Report: Weather Forecast System for Agriculture

Group: 14
Bahattin Tamer Akipek - 22000256
Mehmet Özkan - 22000193
Ahmet Arınç Akyıldız - 22000257
Ali Efe Ekmen - 22331299

Submission date: 1.12.2024
Current semester: 24-25 Fall
Academic supervisor: Prof. Dr. Hadi Işık Aybay

Table of Contents

List of Figures

1. Introduction

Farmers face many problem due to bad weather, like storms, heavy rain or extreme heat which can damage crops, equipment and farm itself. These weather events can cause to big losses if farmers are not prepared. To help solve this, our project is creating a weather prediction and notification system.

This system will check the weather and send farmers useful notifications to help them take action. For example if a thunderstorm is coming, it might send a message like “A thunderstorm is coming. Please check if your lightning rod is working.” Or if there is a chance of frost, it could notify farmers to protect their crops by saying “Low temperature are expected tonight. Cover your crops to avoid frost damage.”

The system will be an easy to use mobile application, giving farmers the information they need at the right time. It will help them reduce risks, protect their work, and make farming more efficient. This report explains the progress make so far in building this system, including its design and key features.

2. Project management plan: Work distributions, meetings

Work Distributions:

Project Manager (Bahattin Tamer Akipek):

1. Reviews and supervises the entire project and ensures that the progress made is completed by the due dates of the tasks.
2. Provides coordination between the teams and establishes communication between them to solve problems.
3. Prepares regular status reports for the team and keeps informed of the progress of the project by reporting the results.

Development Team:

Front-End Developer (Ali Efe Ekmen):

1. Designs, develops the user interface and ensures that it is functional for the end user.
2. Edits designs to ensure compatibility on different devices.
3. Uses programming languages such as React, Angular or HTML when necessary.

Back-End Developer (Mehmet Özkan):

1. Develops the server side and ensures that the system operates and performs.
2. Integrates APIs into the system and manages authentication and authorization processes.
3. Uses technologies such as Node.js, Python or Java.

System Designer (Ahmet Arınç Akyıldız):

1. Creates detailed system architecture and design plans. Works with the development team to ensure that the design meets today's requirements.
2. Manages technical documentation to resolve future system updates and potential problems.

Database Administrator (Ahmet Arınç Akyıldız):

1. Designs the database to support the system's requirements and keeps it up to date.
2. Ensures data integrity for performance, tries to prevent leaks and ensures optimization.
3. Uses tools such as Oracle SQL, MySQL or PLSQL in data management.
4. Prepares backups and recovery plans to prevent data loss.

Analysis and Testing Team:

System Analyst Specialist (Mehmet Özkan):

1. Collects and interprets user needs, ensures that the results are correctly translated into technical specifications.
2. Identifies potential risks or difficulties in the system design and suggests solutions to the team.

Test Specialist (Ali Efe Ekmen):

1. Conducts various testing phases such as unit testing, integration testing and user acceptance testing.
2. Prepares test notes, reports errors and ensures that the system meets quality standards.
3. Uses tools like Selenium, JIRA, or manual testing methods to validate the functionality of the system.

Meetings:

Kick-off Meeting

Purpose: To introduce the project goals, individual responsibilities, and project milestones to team members.

1. To welcome team members and introduce the project’s major milestones.
2. To provide an overview of the project’s goals and delivery timeframe.
3. To assign each team member their roles and inform them of their responsibilities during the project.
4. To determine communication protocols and select the tools to be used and continue with those tools during the process.
5. Q&A session.

Weekly Development Meetings

Purpose: To review the project’s development progress, resolve issues, and plan upcoming tasks.

1. To get updates from everyone on their current tasks and see how the processes are progressing.
2. To review completed tasks and review the current status of features.
3. To ensure coordination between front-end, back-end, and database tasks and discuss solution

strategies if there are any issues.
4. To plan tasks for the upcoming week and ensure that any areas that require support are resolved through coordination.

Monthly System Design Meetings

Purpose: To ensure that the design is consistent with user requirements and technical constraints.

1. Review the system design and architecture, and ensure that the design is improved and move on to the next steps.
2. Determine user requirements, integrate them into the existing system, and discuss their implementation.
3. Address any technical challenges or constraints that may arise.
4. Make new updates to the technical documentation.

Test Feedback Meetings

Purpose: Address issues that arise during the testing phase and ensure quality standards.

1. Share test results with the manager and other team members.
2. Discuss identified bugs and their impacts, and create a new roadmap and plan for fixing them.
3. Prioritize bug fixes and feature adjustments.
4. Review verification of fixes made in previous testing sessions.

Final Project Review

Purpose: Summarize results, collect team feedback, and document lessons learned for future projects.

1. Overview project deliverables and results.
2. Review team performance and collaboration.
3. Gather feedback from each team member and take a general look at the process.
4. Identify successes and areas for improvement.
5. Document mistakes and lessons learned for future projects.
6. Give closing remarks and recognize the team.

3. Information Collection Method: Interviews and surveys

Interviews and surveys were conducted as our methods of gathering information to develop the weather forecast and notification system. As a result of this research, we analyzed the methods farmers prefer in agriculture and the challenges they face, which helped us understand the requirements for our system

Interviews: Conducted face-to-face or online with farmers, agricultural engineers and weather workers.

1. Weather-related challenges farmers face.
2. How farmers are currently preparing for and responding to extreme weather conditions.
3. Any useful feedback or advice.
4. And their preferences for receiving notifications were investigated.
5. Interviews were completed in one-on-one interviews and questions that were not included in the survey were asked during the conversation.

Surveys: The aim was to analyze a wider group of farmers in different regions, access was provided via internet, and for those without internet infrastructure, access was provided through paper-based surveys.

1. Frequency and severity of weather-related problems in farming. Level of technology adoption among farmers.
2. Common crops, equipment and farm types that require protection from weather events.
3. Application features and usability preferences

By combining data from interviews with data from surveys, we reached a main conclusion, which allowed us to adjust the design and functionality of the system to effectively address the practical challenges faced by farmers and make a more effective project.

4. Analysis of Similar Systems

FarmLogs

- Functionality: It provides tools for managing agricultural activities by integrating weather prediction data with crop health. It improves operational planning and product management.
- Technology: Generates weather predictions and provides recommendations for crop health by using weather APIs and satellite data.
- Focus: Useful for farmers needing for a system combining weather forecasts with agricultural advice.

Climavision

- Functionality: Provides hyperlocal weather forecasts by using AI models developed considering agricultural purposes. Provides forecasts from hyperlocal weather to global forecasts, point-specific forecasts with high resolution predictions.
- Technology: Uses large observational datasets, numerical weather predictions and AI to maximize precision.
- Focus: Ideal for risk management and planning in agriculture.

AccuWeather

- Functionality: Provides local weather predictions and alerts.
- Focus: It is a general weather prediction system that can be used in agriculture as it provides alerts for extreme weather conditions.

ICT-Based Systems for agriculture.

- Functionality: Uses mobile tools to transmit weather information. It provides SMS alerts and voice messages.
- Focus: Provides services for smallholder farmers with limited technology access.

5. Strengths and Weaknesses of the Similar Systems

Strengths:

- Accuracy: High resolution forecasting models such as Horizon AI which is used by Climavision provide point-specific weather data that help farmers to handle daily agricultural operations.
- Risk reduction: Many of existing systems provide early intervention to prevent possible losses from extreme weather conditions such as droughts, floods, storms etc.
- Ease of use: Especially ICT (information and communication technology) based systems are easy to use as they do not require any technical information.
- Value-added features: Existing systems integrate weather forecasts with many other features such as crop management, irrigation planning etc.

Weaknesses:

- Infrastructure problems: Most of the systems require a stable internet connection or advanced sensors which may be unavailable in many rural areas.
- Implementation costs: As these systems implement advanced technologies including machine learning, complex sensors etc, and R&D costs are extremely high, they are not affordable for some smallholder farmers.
- Generalization in models: Some system built on global models, may lack details required for variable local climates.
- Complexity: Advanced systems require technical training for setup and effective use, which makes adoption difficult for less developed areas.

6. Proposed Structure of your system

Machine Learning:

We have chosen LSTM (Long Short-Term Memory) algorithm as it is most appropriate

option for processing data broken into timestamps. To implement this algorithm, we are going to use TensorFlow and PyTorch libraries provided by Python. The model takes; historical weather data, OpenWeather forecasts and real-time sensor data as input and puts forward predictions for 7 days.

Data Storage:

We decided to use MySQL to store all data coming from sensors, OpenWeather API and AI model. It is a relational database system that provides an environment to store related data. It would be more efficient to process the large data set, when some relationships between obtained data are established.

Data Processing:

In this layer, external data coming from sensors and OpenWeather API goes through some processes and rearranged to a unified form in order to reach appropriate form to be processed by the AI model. Missing or erroneous data is handled in this layer. For this data manipulations, some libraries such as Pandas and NumPy built in Python, are going to be used.

Notification Triggering:

Based on detected weather conditions, if there is a situation that it must be intervened immediately, notifications are triggered in this layer. For app notifications, OneSignal API is implemented. And according to users preferences, SMS messages are sent by Twilio SMS service.

External Data Sources:

Real time data like humidity, temperature, wind etc are obtained from the sensors and sent to the data processing unit. Furthermore, historical weather data and some forecasts are needed as well as real time data. To obtain this information we use OpenWeather API.

User Interface:

A mobile application that provides an interface for farmers to get alerts, notifications about the situation of the crops and advice, and set user preferences, for example, in some rural areas, internet infrastructure may not be stable, so user have the chance to receive SMS messages. The graphical interface is going to be developed using react which is an up-to-date framework providing many beneficial tools for creating a modern and easy to use interface.

7. High level Architecture diagram, Use Case Diagram, Organization scheme (Some example diagrams for other unrelated projects are given below)

Figure 1: *Organization scheme*

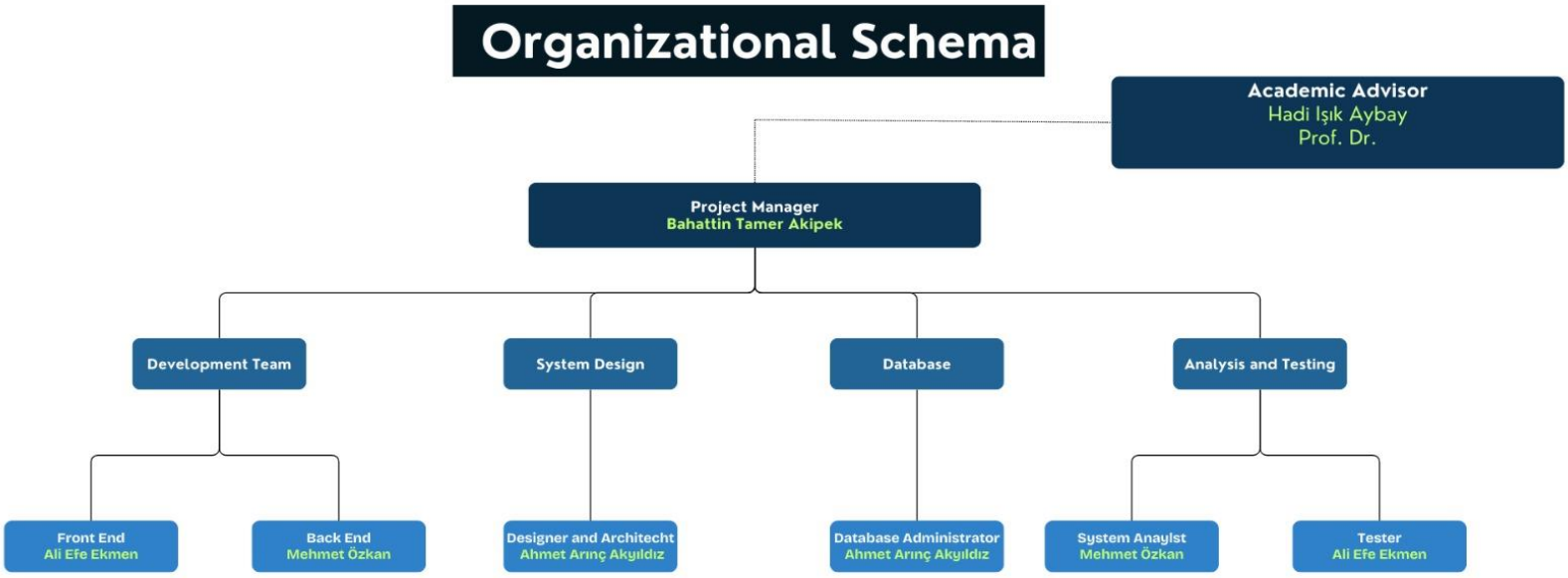


Figure 2: *High level architecture diagram*

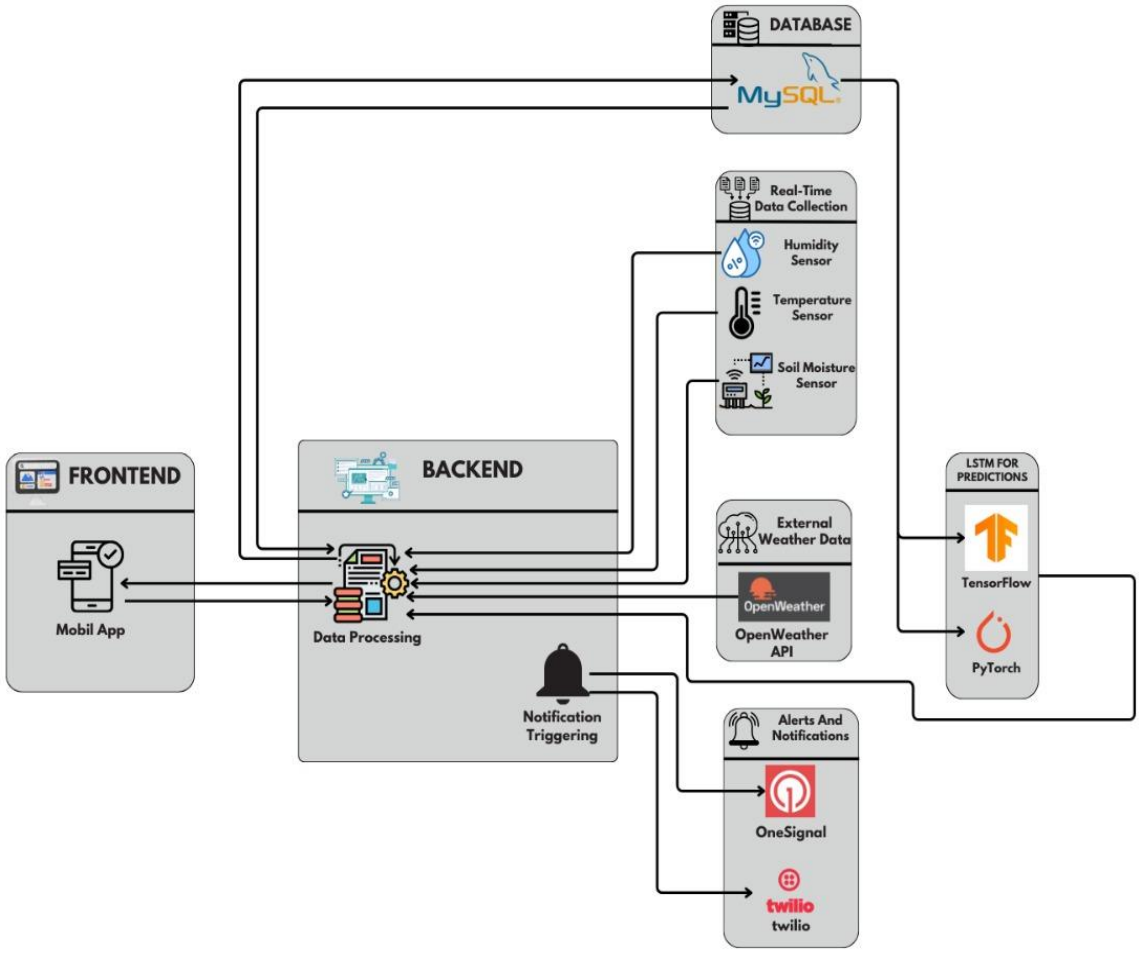
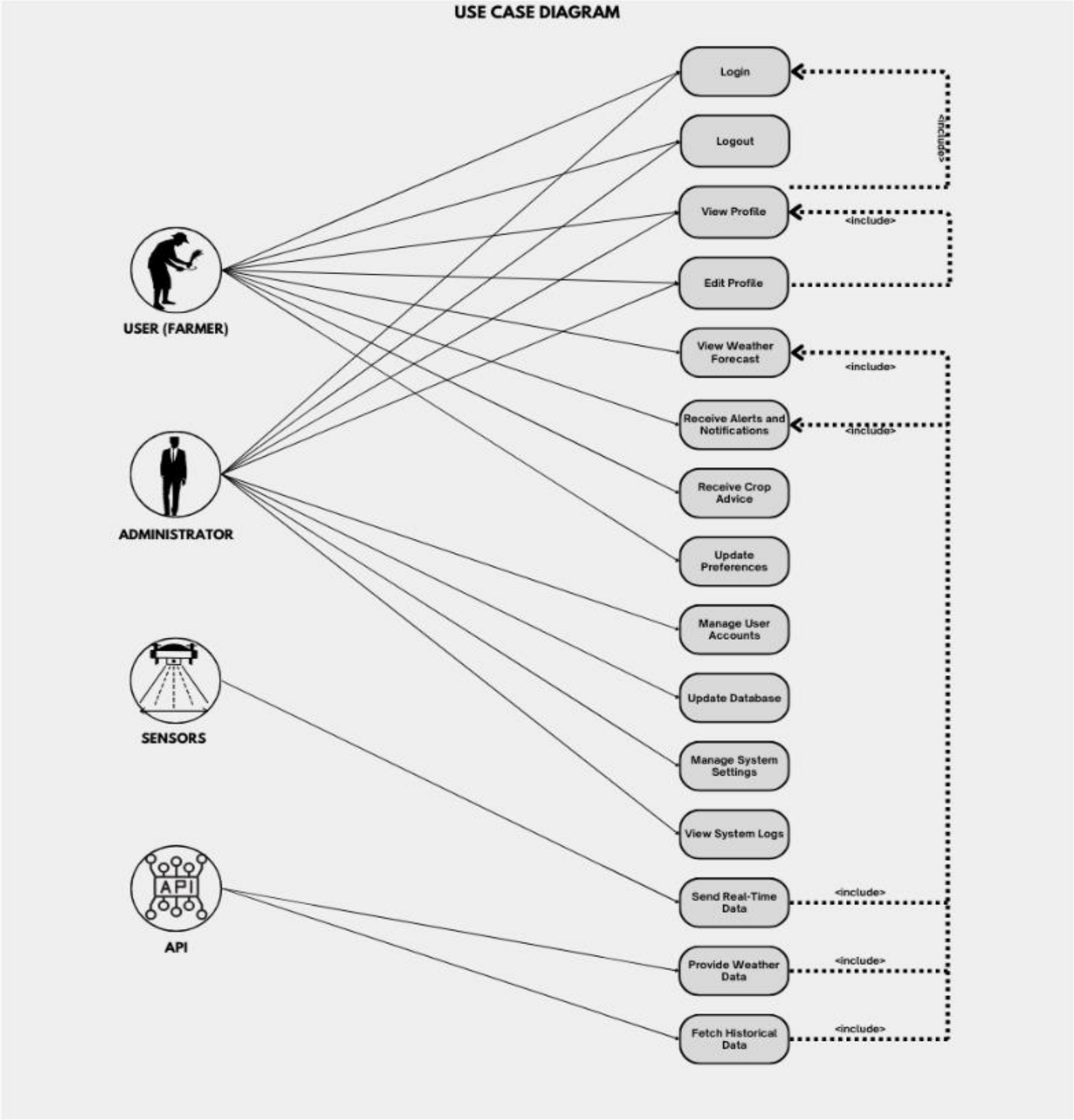


Figure 3: *Use case diagram*



8. Conclusion

This system is made to help farmers protect their crops and equipment from bad weather. By giving early warnings and useful advice, the system will help farmers prepare for things like storms, high temperatures and heavy rain, reducing damage and losses.

So far, good progress has been made in creating the system. It can check weather data and send helpful notifications. But more work is needed to make it more accurate, easy to use and test it with real farmers.

In next steps, we will focus on improving weather predictions, making the app simple to use, and making sure all farmers can access it. When it finished, this system will be a useful tool to support farmers and make farming easier and safer.

9. References

- https://www.farmintell.com/blog/the-impact-of-weather-forecasting-on-modern-farming/
- https://climavision.com/agriculture/
- https://www.farm-d.org/app/uploads/2019/05/Using-ICT-to-Provide-Weather-Information-for-Agriculture.pdf
- https://bureauofmeteorology.org/blogs/weather-science/the-impact-of-weather-on-agriculture-a-comprehensive-anal
- https://openweathermap.org/
- https://www.visualcrossing.com/agriculture-weather-data
- https://deepmind.google/discover/blog/graphcast-ai-model-for-faster-and-more-accurate-global-weather-forecasting/
- https://www.localline.co/blog/best-apps-for-farmers-and-agriculture