MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER ENGINEERING DESIGN

SOFTWARE DESIGN OVERVIEW

AGROINTELLIGENCE PROJECT

Project Supervisor

Asst. Prof. Selim Temizer

Project Team Members

Fatih ACUN

İsmail BALABAN

Onur Yiğit ARPALI

Furkan MURAT

TABLE OF CONTENTS

1. Product Description 3
2. High Level System View 4
3. Overall Design 5
3.1. Design Rationale 9
3.2. Alternative Design Options 9

TABLE OF FIGURES

Figure-1. Context Diagram 4
Figure-2. Deployment Diagram 5
Figure-3. Component Diagram 7
Figure-4. ER Diagram 8

1. Product Description

Sunn pest is a well-known wheat pest that can be observed in the Middle Eastern Region. The Sunn pest has three main phases on its life-cycle. It spends the first phase in the 'kışlak'. In the second phase, it travels from 'kışlak' to wheat field. Finally, phase three corresponds the time that the Sunn pest spends in the wheat field. We can also divide phase three in to different stages according to the Sunn pest's evolution steps. During phase three, the wheat field should be sprayed at a certain step to minimize the damage in the field.

The current solution for determining the pesticide application time is done by simply counting the pests in the wheat fields by agricultural engineers. This method is time consuming and inefficient. It takes nearly 3 months of an agricultural engineer in wheat fields of different regions. That is also costly for Ministry of Agriculture.

Our product "Development of a Forecasting and Warning System on the Ecological Life-Cycle of Sunn Pest" aims to predict the pesticide application time of the Sunn pest. The product will detect the phase of the Sunn pest by applying machine learning algorithms on the climate data and label data provided by Ministry of Agriculture. After detecting the time when the Sunn pest is in phase three, then the product will detect the time when the Sunn pest becomes most harmful to the wheat field and warn the farmers that the pesticide application time comes.

The main components of the project are listed below:

- Data collection software that pulls the climate data from METOS web services.
- Data preprocessing software that makes the data suitable for ML models
- Prediction Models that estimate the phase and pesticide application time.
- Triggering Software
- MySQL Database
- Web Page
- SMS Warning System

2. High Level System View

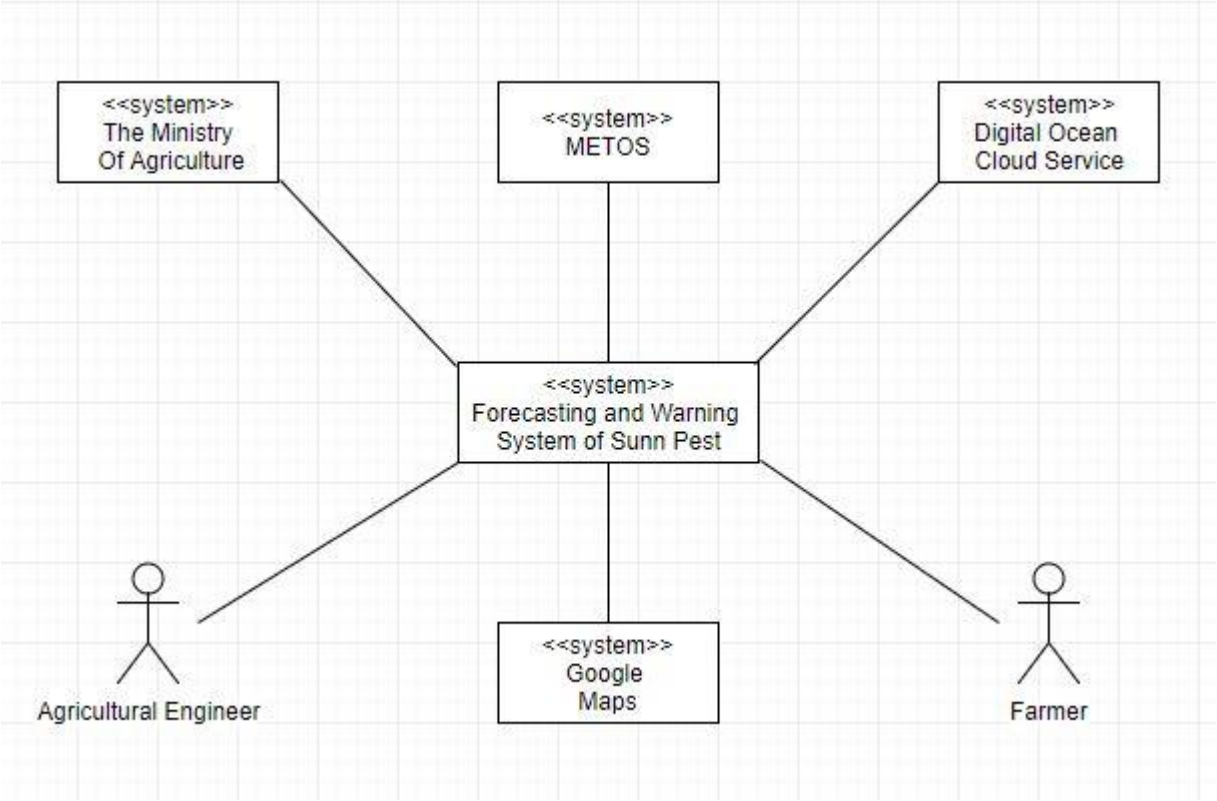

Figure-1. Context Diagram

Users interacts with the system using interface via web browsers. Interface will be implemented as HTML pages.

Digital Ocean will be used as cloud service. Our main server is working on this cloud server and it can be connected globally.

Google Maps is the API provider for the map in our webpage. We are using this API to show the locations of our stations.

Metos is the manufacturer of the weather stations that we are using. Metos also provides web services so that we can fetch the climate data and place that data to our database.

This project will be used within The Ministry of Agriculture. The Ministry also provides label information and supports the project by sharing field specific information about Sunn Pest.

3. Overall Design

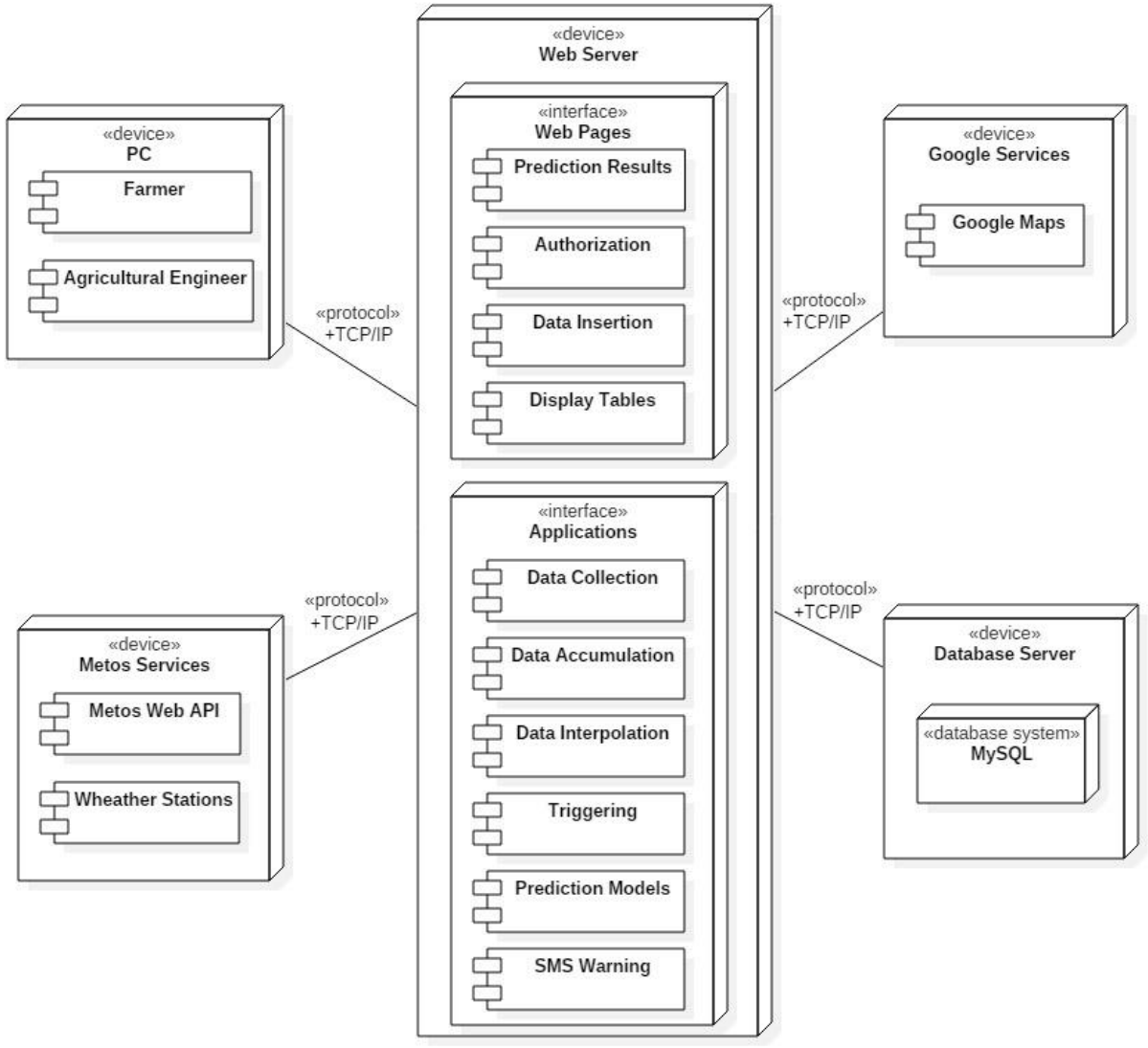

Figure-2. Deployment Diagram

You can see the deployment diagram in Figure-2.

On the client side, farmers and agricultural engineers will be our user agents. Their role will be described in detail when explaining the class diagram. They will interact with the Web Server that you can see in the middle.

Web server consists of Web Pages and Applications that runs on it. There will be four main web pages. Their roles are stated as follows:

- **Prediction Results Page:** In this page farmers and engineers can see the prediction results of different fields. There is a dropdown menu and a map to select the field. After selecting the user can see the results in a tabular format. You can reach the page from 46.101.110.16.
- **Authorization Page:** This page is for the users that have right to access the admin panel. that consists of 'Data Insertion' and 'Display Tables' pages.
- **Data Insertion Page:** In this page farmers and agricultural engineers can insert the label data that will be used in the prediction models as train set.
- **Display Tables:** This page is for the Agricultural engineers to see the database tables (Climate Data, Station Data etc.)

There are six main applications that runs on the server. These applications are the core programs that are used to keep the system alive and functional. Their roles are stated as follows:

- **Data Collection Software:** Pulls data from the web services run by METOS weather stations.
- **Data Accumulation Software:** Accumulates the climate data in order the use accumulated data in prediction models.
- **Data Interpolation Software:** Interpolates the NULL climate values due to the malfunction in the weather stations.
- **Prediction Models:** Machine Learning Models that are built for phase and pesticide application time.
- **Triggering Software:** Triggers the Data Collection Software, Data Accumulation Software, Data Interpolation Software, Prediction Models respectively every day. CRON scheduler is used for this task.
- **SMS Warning System:** Sends SMS to the farmers just before the pesticide application comes.

There are three services that we used in the project. Their usages are stated below:

- **Database Server:** Stores the climate data, label data, station data etc.
- **Google Services:** Provides a Map API that is used to choose field locations in the prediction results page.
- **Metos Services:** Provides a web service to allow us to pull the climate data.

In the figure below (Figure-3) you can see the component diagram. This diagram is provided to visualize interactions of components with each other.

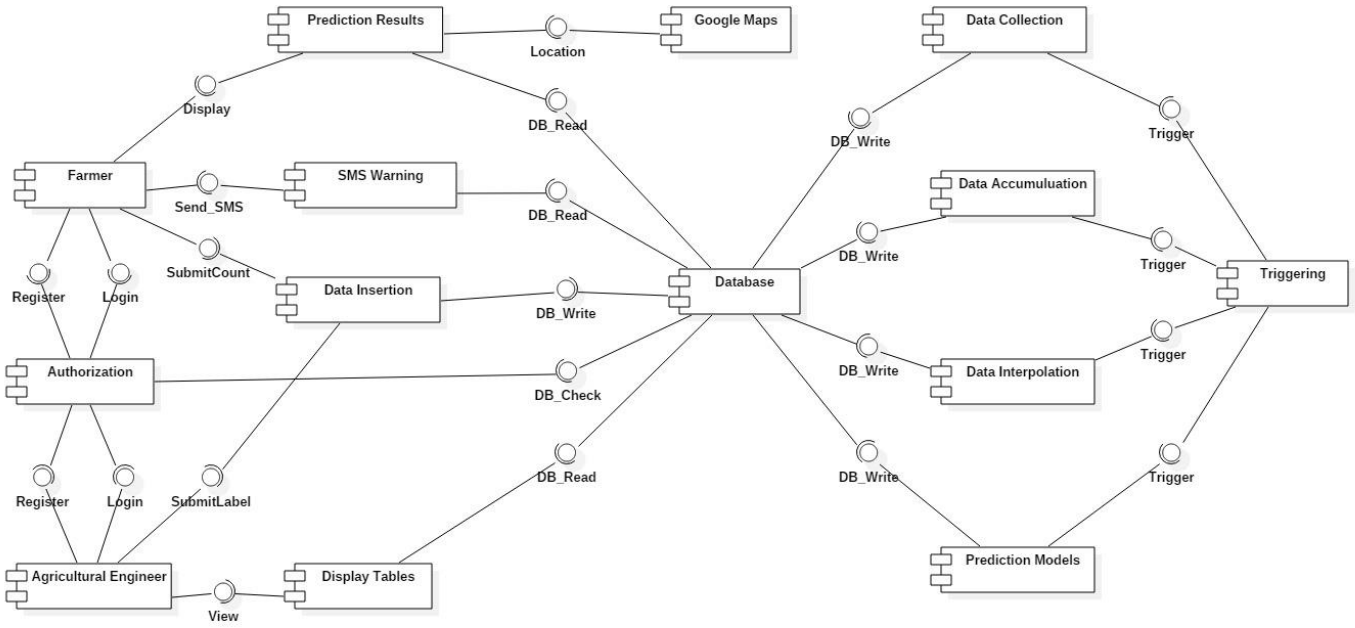

Figure-3. Component Diagram

The components and their roles are described below. Some of them is not explained in order not to replicate the previous information given above.

- **Farmer:** Farmers will interact with 'Prediction Results' and 'Data Insertion' web pages. From the Data Insertion page, farmers submit the Sunn pest count in the fields. They should have Login to connect the Data Insertion page. This action is accomplished by Authorization component.
- **Agricultural Engineer:** Engineers insert LabelData via the 'Data Insertion' component. They also have the right of displaying database tables via 'Display Tables' component. They must authorize via 'Authorization' in order to use these components.
- **SMS Warning:** This component is used to warn the farmers with SMS.
- **Google Maps:** Provides 'map' to the prediction results page.

In the figure below (Figure-4) you can see the ER diagram.

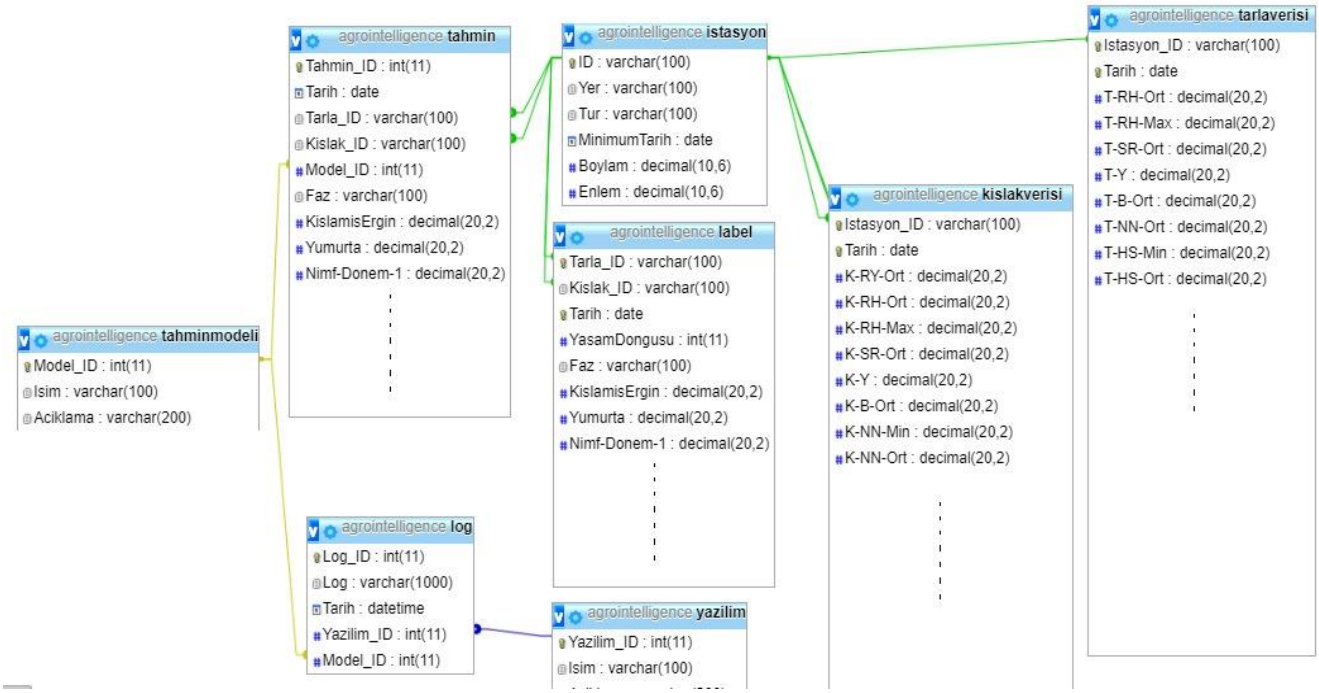

Figure-4. ER Diagram

The explanation of tables is listed below:

- **'istasyon' table**: Stores the weather station's ID, place, type etc.
- **'tarlaverisi' and 'kıslakverisi' tables:** Store the climate data. istasyon_ID is the foreign key that references istasyon table.
- **'tahmin' table:** Stores the prediction results of the prediction models. Tarla_ID and Kislak_ID are foreign keys that references the 'istasyon' table. Model_ID is the foreign key references 'tahminmodeli' table.
- **'label' table:** Stores the labels that are used by prediction models. Tarla_ID and Kislak_ID are foreign keys that references the 'istasyon' table.
- **'tahmin modeli' table:** Stores the Model_ID, name and description of prediction models.
- **'log' table:** Stores the logs of the software components and prediction models. Every software fills that table after each operation so that we can keep track of the operations of them. Yazılım_ID is a foreign key that references 'yazilim' table. Model_ID is a foreign key that references 'tahminmodeli' table.

3.1. Design Rationale

We find suitable to use object oriented approach for the applications. The reason for choosing object oriented approach is make it possible to exploit divide and conquer method for simplifying the project development process. In addition, this approach helps us to separate objects and methods. We use Java(SE7) to implement the Data Collection, Data Accumulation, Data Interpolation, and Prediction software. We also used JPA to map the database tables into Java objects.

For the model development we use WEKA which is an open source project.

For the web pages, we use JavaScript, HTML, CSS, PHP and Bootstrap.

3.2. Alternative Design Options

For the model development part, we used J48 trees and random trees. The reason for this choice is the relatively small amount of data that we have. Because these models are linear, we avoid the problem of over-fitting. If we have used an algorithm like Artificial Neural Networks, it will perfectly classify the train set but it will fail when the real test data is provided to it.

For the applications that run on the server we used Java. These applications can be run by Python, C++ etc. The reason for choosing java is to exploit JPA (framework that maps database tables to java objects). Also, the machine learning tool WEKA that we are using provides a JAVA API. As a result, we decided to use JAVA as a programming language.