TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE491 - PROJECT ANALYSIS REPORT

Project Title

ALSApp (Agriculture and Livestock Support Application)

Project Supervisor

Venera ADANOVA

Project Juries

Ayşe Yasemin Seydim

Tansel Dökeroğlu

Project Members

Engin Samet Dede

Sevgi Dilay Demirci

Gizem Ünsal

22/11/2024

Table of Contents

1. Introduction

1.1 Project Description

The goal of our project is to develop an application that will make tracking easier for farmers in a variety of ways. Furthermore, this platform combines all the necessary resources to improve livestock performance and farming activities. Users will be able to enter their animals' health data and receive relevant information, track their crops digitally for optimal efficiency, and track the growth and development of their crops and animals in real time using an easy-to-use interface that is favourable to both farmers and livestock.

By consolidating all agricultural and livestock management needs onto a single platform, this system enables users to manage their operations in a more organized, efficient, and planned manner. It also provides instant updates on industry developments, helping users stay informed and responsive to industry changes. They will be aware of government support, mainly offered by the Ministry of Agriculture and Forestry, and benefit from funding opportunities. In addition, the following process can also be used by the Ministry of Agriculture and Forestry staff to keep track of the reliability and accuracy of support.

1.2 Purpose

This documentation gives a guideline and expanded analysis on creating an application for agriculture and livestock support. Moreover, it points out the system's capabilities and features, which will have been made when the project is finished; along with these, it also mentions interfaces and how the application will respond to possible external inputs. Besides, the documentation extends its coverage with technical specifications and resources crucial to creating and maintaining the application.

On the other hand, this document covers a simple general aspect of the development procedure, project management methodology, the developer team's roles and responsibilities, and the project timetable. The documentation also refers to the potential risks and challenges that can appear during the application's development, as well as the solutions that will be planned to eradicate these problems.

1.3 Document Conventions

The standards and typographical conventions can be utilized during the development period of the Analysis Report for the ALSApp (Agriculture and Livestock Support Application), which has been stated below:

1. Each requirement statement's priority level will be determined. The priority level will show the importance of the statement's successful termination based on its priority, and accordingly, project success will increase. For example, the priority of reaching the Ministry's support and tracking the data for the clients' products is high, and such statements will be considered initially.

2. The Analysis Report should satisfy the capability for readability and clarity. Therefore, it will utilize a noncontradictory typeface and formatting design until the end of this report. Moreover, it will provide easy-to-read documentation as much as possible on the used device.

3. The Analysis Report will acquire a dictionary of terminology to ensure the terminological integrity and rationality of the documentation.

Using these principles, the application will be easy to develop with a decreased difficulty level, using the priority of the statements.

1.4 Intended Audience and Reading Suggestions

This documentation has been created for us as the developer students for our senior project. It will provide detailed information about the application with the required analysis to the project supervisor and juries. Moreover, the Ministry of Agriculture and Forestry will also be able to access this documentation when they use the project website. It is recommended to read it by following the suggestions given throughout the report

1.5 Product Scope

This project will use technology to fortify the tie between farmers, livestock breeders, and the Ministry of Agriculture and Forestry. The application offers clients a comfortable and user-friendly platform that they can use to track their products, weather, and the support of the Ministry. The clients will be able to utilize the app efficiently to navigate through the support of the Ministry, track their products continuously, and help them be updated on the possible weather outcomes.

On the other hand, this app will increase awareness of the government's support since it takes a lot of work for clients to reach them. The effortless usage of the app will boost knowledge of the support and can benefit the clients and the Ministry mutually.

2. Proposed System

2.1 Overview

The ALSApp will be created for the benefit of farmers and livestock breeders. It also aims to enable the Ministry of Agriculture and Forestry and its employees to track existing aid. With this application, the main problems that farmers and livestock breeders generally experience will be taken into consideration, as will the Ministry of Agriculture and Forestry and its employees tracking their assistance. This will be explained in detail below in this section.

Farmers and livestock breeders can track their products and resources: This application aims to create a portal where farmers and livestock breeders can record their product and resource data in the application and track them in detail. With this facility, as products and resources are tracked, possible incomplete harvests or low livestock yields can be detected, and early intervention can be made to prevent more significant problems. Moreover, this will increase productivity and make it easier for farmers to achieve more sustainable and economical agriculture and livestock farming since problems can be overcome early.

Facilitating access to Ministry supports: The Ministry of Agriculture and Forestry currently offers various support for farmers and livestock breeders. However, since these supports are

available on their own website, and it is tough for farmers to research them due to low digital literacy, this application offers farmers a more accessible option to access these supports.

Weather monitoring: Weather monitoring is very critical for farmers and livestock breeders. A possible weather disaster or excessive rainfall can cause severe problems. This application aims to ensure that farmers are informed about the weather much earlier.

Access for the Ministry and its employees to check the support: This application aims to facilitate the checking and detection of errors and deficiencies for the employees working in the units related to support in the Ministry. In this way, the Ministry employees will be able to notice and solve possible deficiencies and errors in the supports in a shorter time.

The application will be created with these purposes in mind, and it aims to minimize the existing deficiencies of both parties. Thanks to the application, farmers, livestock breeders, and the Ministry will benefit from increased benefits and digital convenience.

2.2 Functional Requirements

The system's functional requirements are described here, but it is recommended that the Scenarios section is first reviewed to understand the functional requirements more quickly. (Part 2.5)

2.2.1 User Registration and Profile Management:

Description and Priorities

The user will register to the application. When the necessary information is entered, the user will have an account of their own in the application. If this registration process has been done before, the user will be able to log in to the application with their existing account.

Priority Level: High

Functional Requirements

FUNC-REQ-1: The system shall register a new account by requesting users' information, such as their first name, last name, phone number, and password.

FUNC-REQ-2: The system shall check the user's input to ensure that all the required fields are filled out and valid.

FUNC-REQ-3: The system shall give an error message if any inputs are invalid.

FUNC-REQ-3: The system shall collect the user's data, save it in a safe database, and create a unique user ID.

FUNC-REQ-4: The system shall allow registered users to log in with their phone numbers and passwords.

FUNC-REQ-5: The system shall check that the user's phone number and password are correct and correspond to the information in the database.

FUNC-REQ-6: The system shall display error messages if the user's input is incorrect or the login fails for any other reason.

FUNC-REQ-7: The system shall allow users to set their password back to another if they cannot remember it using a phone number.

FUNC-REQ-8: The system shall send users a unique password reset link to their e-mail address.

FUNC-REQ-9: The system shall test the validity of the password reset link and permit the user to input the new password.

FUNC-REQ-10: The system shall assign each user a unique User ID and safely store all their data in the database.

FUNC-REQ-11: The system shall allow users to change their passwords.

2.2.2 Data Entry for Crop and Livestock Tracking:

Description and Priorities

The system should enable the user to enter and follow the data of crops and animals. If there is any change, such as a decrease in wheat production, the user should be warned about this change.

Priority Level: High

Functional Requirements

FUNC-REQ-1: In addition to permitting persons to enter new product information, the system shall allow them to click the “Enter Product” button to start the data entry process.

FUNC-REQ-2: The user shall see a window where they can input the product name and characteristics by clicking the “Enter Product” button.

FUNC-REQ-3: The system shall check the data entered by the user in the product form; thus, all necessary fields must be correctly filled out.

FUNC-REQ-4: If data is missing or contains factual errors, the system shall send a plain text message describing what should be filled in or corrected.

FUNC-REQ-5: The system shall save product data to the database only if data validation is successful to the user when the form submission has been completed.

FUNC-REQ-6: The system shall store the usual data entries in the database, explaining and covering the animal tracking and routine status data.

FUNC-REQ-7: The system shall then perform statistical data analysis in order to foresee situations that are likely to be discovered in the future or unfavorable trends.

FUNC-REQ-8: The system shall evaluate data changes that occur over time to determine whether they reveal a potential hazard or have a negative impact.

FUNC-REQ-9: If the information analysis determines a dangerous scenario or a negative impact, the system shall notify the user and advise them on possible risk management.

2.2.3 Weather Alerts and Notifications:

Description and Priorities

The weather monitoring interface for farmers and livestock breeders aims to provide instant, precise, and reliable weather monitoring. For this, weather information will be updated using instant information obtained from the internet. Users will be notified of possible hazards or weather events according to their importance.

Priority Level: High

Functional Requirements

FUNC-REQ-1: The system shall get and show the user real-time, accurate weather information from the internet.

FUNC-REQ-2: The system shall allow the user to check the weather conditions by touching the “Weather” button on the interface.

FUNC-REQ-3: Instead of showing the weather window at first, the “Weather” button will navigate the user to an easy-to-use interface where the current weather conditions can be seen.

FUNC-REQ-4: If the weather is unsafe or inappropriate, the system shall alert the user as a preventative measure.

FUNC-REQ-5: The system shall provide the user with information on what environmental conditions, such as humidity and temperature, are suitable for activities like planting or other sensitive atmospheric phenomena and even diseases.

2.2.4 Access to ministry supports:

Description and Priorities

The application aims to use a convenient system for displaying support to facilitate access to the Ministry’s support. The data will be taken directly from the Ministry of Agriculture and Forestry and will be added to the database of ALSApp immediately; throughout this platform, livestock breeders and ministry employees will be able to be aware of the possible supports by using this feature to access support more easily since it’s even hard for the ministry employees to be aware of the support that they are giving because of the current complicated system.

Priority Level: High

Functional Requirements

FUNC-REQ-1: The system shall draw real-time support information from the Ministry’s website.

FUNC-REQ-2: The system shall locally store updated support information so that users can easily access it.

FUNC-REQ-3: The system shall allow the ministry employee logged into the application to mark a specific support item as an issue by pressing a “Flag” button if an error exists in that support.

FUNC-REQ-4: The system shall send a notification to a user each time the Ministry submits new support information.

2.3 Non-Functional Requirements

2.3.1 Performance Requirements

The system's performance is a critical factor in its availability. The system's performance in retrieving information, saving data, and sending notifications is crucial to the application's efficiency.

2.3.2 Reliability Requirements

The data and information in the application must be reliable for the user. The reliability of these factors ensures that the user and the Ministry are satisfied with the application during its use.

2.3.3 Software Quality Attributes

This application shall have a system that provides ease of use and has a practical and user-friendly user interface. Users should have no difficulty navigating the user interfaces and be able to fill in the forms easily and quickly.

2.3.4 Usability Requirements

This application considers the users' digital literacy and aims to prevent them from experiencing difficulties during use. From the users' point of view, this application will be user-friendly, and user interfaces will be designed to minimize their challenges.

2.3.5 Security Requirements

The safety of users will be taken into consideration when designing this application. Since the application will already contain users' personal data, it will be developed in accordance with the necessary security regulations. The security of the users' accounts will be protected from threats with a two-factor verification system sent to the e-mail or phone number.

2.3.6 Scalability Requirements

Since the application will be used by many users at the same time, the simultaneous use of these users should be smooth in the use of the application. The user should not encounter usage difficulties (bugs, slowness, etc.) in such cases.

2.3.7 Maintainability Requirements

The system should be modular and not cause difficulties for software developers when it comes to maintenance, such as system bug fixes for problems that may be encountered in the future. This modularity is expected to facilitate maintenance on particular parts of the system.

2.4 Pseudo Requirements

2.4.1 Platform Compatibility
The system should be suitable for use in operating systems such as Android, IOS and have high compatibility with these operating systems.

2.4.2 Language Support
The system should provide multiple different languages in the user interface to its users.

2.4.3 API Integration
The system should be able to get the real time updates. Hence, it should be secured using APIs.

2.4.4 Performance Metrics
The programs response time should not be long. Therefore, the response time is expected to be approximately about 2 seconds.

2.4.5 Coding Languages
The application should be developed using the required different coding languages for better functionality.

2.5 System Models

2.5.1 Scenarios

2.5.1.1 User Registration and Profile Management:

Stimulus/Response Sequences

1. In order to register for the application, the user will first click the “register” button.

● The system will generate and display a registration form on the screen.

● The system will verify the information entered by the user and will give an error message if there is a problem.

● The system will create a new account for the user and keep the information of this account in its database.

● Finally, the system will direct the user to the login page.

2. After entering their mobile phone or e-mail address and password on the login screen, the user presses the “log in” button.

●The system verifies the information entered by the user, and if the user enters incorrect information, the system gives an error message.

● The system checks the user database to find the user’s information.

● If the user’s information is found, the system allows the user to log into the application successfully.

● If the system cannot find the user’s information in the database, it gives an error message and does not allow login.

3. The user taps the “Forgot Password” button.

●When the password is forgotten and the “I forgot my password” button is pressed, the system will direct the user to a form asking for their phone number or e-mail address.

●The system will display an error message if this information is entered incorrectly.

● The system generates a password reset link and sends it to the user’s e-mail or phone number.

● The system displays a message that the password reset link has been sent to the user’s e-mail or phone number.

4. The user clicks on the password reset link in their e-mail.

● The system verifies the validity of the password reset link and displays a form for the user to enter their new password.

● The system validates the password and displays an error message if necessary.

● The system updates the user password in the database and displays a success message.

● The system redirects users to the login page to log in with their new password.

2.5.1.2 Data Entry for Crop and Livestock Tracking:

Stimulus/Response Sequences

1. Data entry into the system

●For system data entry, the user must click on the ‘enter product’ button to enter the name and properties of the products.

●When this button is clicked, the system opens the interface for data entry.

●The system saves the data to the database when the user fills in the form. It will give an error message in case of any missing or incorrect input.

2. Tracking the Products and Animals

●As the user enters regular data into the system, it will be recorded. Thanks to these regularly entered data, the follow-up of the products and the routine status of the animals are accurately recorded.

● The system can predict possible future dangers and negativities increases thanks to the recorded data.

3. Notify the user about the possible negative outcome

●The system takes into account the changes in the entered data and determines whether the change is harmful or not.

●If the system encounters a significant hazard or change, the user should be notified with notifications.

2.5.1.3 Weather Alerts and Notifications:

Stimulus/Response Sequences

1.Instant weather information extraction:

●The system will provide instant, precise, and accurate information to the user by retrieving weather information from the internet.

2.Users will be able to learn about the weather conditions when in need:

●The system will allow users to enter weather information when the 'weather' button is pressed.

●The system will redirect the user to an easy-to-use user interface where the user can follow the instant weather conditions when the button is touched.

3.Users will be notified about the weather conditions that they need to be cautious about

● The system should send a notification to the user in case of any problematic weather event.

●Based on the humidity and weather temperature, the system should notify users that it is a good time for planting or other useful activities.

2.5.1.4 Access to ministry supports:

Stimulus/Response Sequences

1.Reaching the Supports:

●The system retrieves support information instantly from the Ministry's website.

●The system saves this information on the application and makes it available in the application.

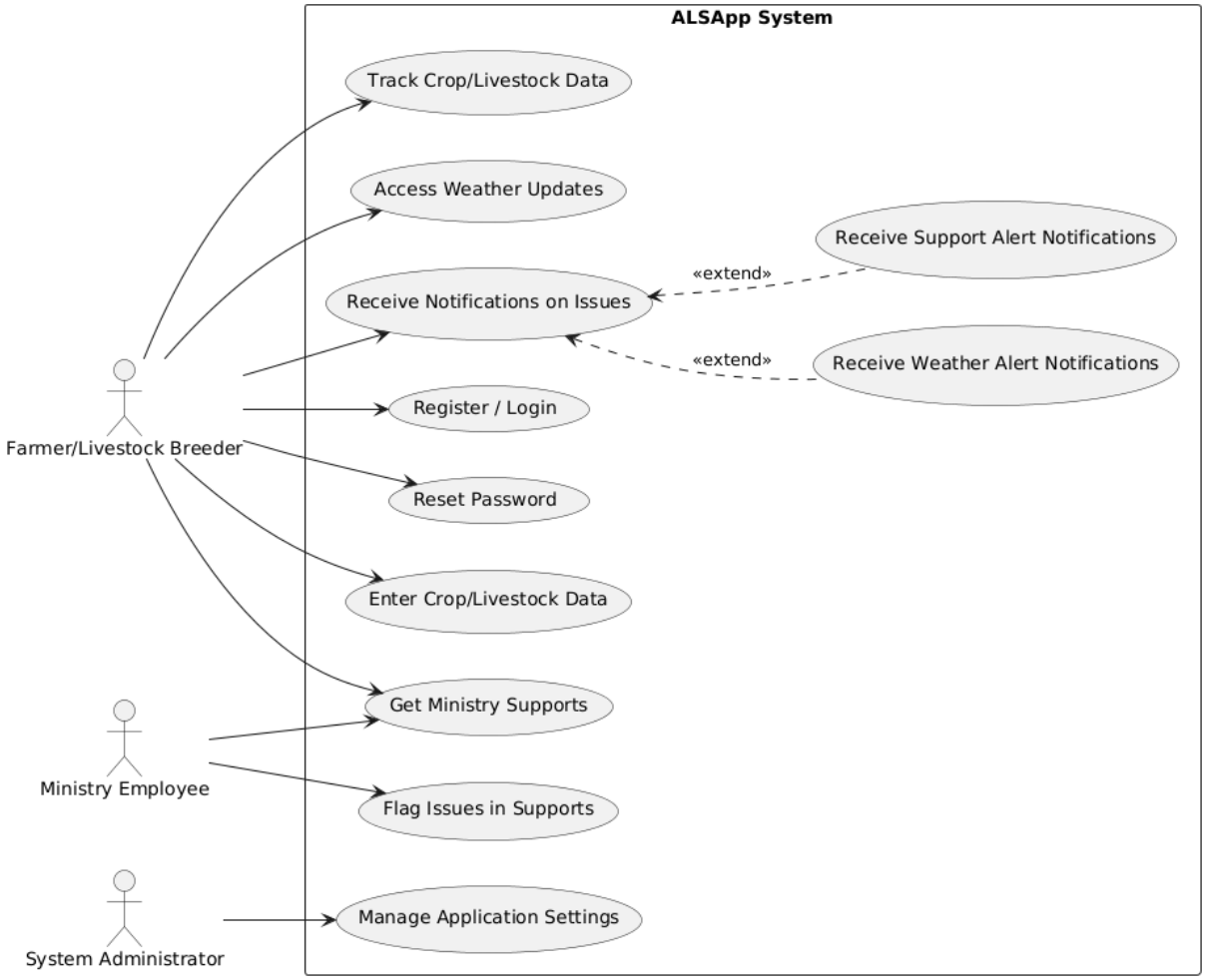
2.Flagging the support:

●The system should have the ability to flag that support by pressing the 'flag' button if there is a problem in the support for a user logged in as a ministry employee.

3.Sending notification:

●The system sends a notification to the user for each new support from the Ministry.

2.5.2 Use Case Model



Actors

Farmer/Livestock Breeder:

The main user of the system interacts with almost all the features of the system to manage agriculture and livestock data and livestock data and stay informed about weather and ministry support updates.

Ministry Employee:

A government employee who is in charge of monitoring and managing the issues regarding support to agriculture and its related livestock.

System Administrator:

This actor is responsible for managing the system's settings and maintaining its overall functionality.

Use Cases
Track Crop/Livestock Data

* Actor: Farmer/Livestock Breeder

* Description: Enables the user to track the progress of their crops and livestock in terms of their growth and health. Tracking data is very critical as it helps the farmer make informed decisions to enhance agricultural productivity.

Access Weather Updates

* Actor: Farmer/Livestock Breeder

* Description: Allows the user access to information about the current weather to take necessary precautions that could affect farming and livestock.

Receive Notifications on Issues

* Actor: Farmer/Livestock Breeder

* Description: Informs the users about critical issues like hazardous weather or support-related alerts by sending them notifications.

* **Extensions:**

    * **Receive Weather Alert Notifications:** Extends the base notification functionality to provide alerts for adverse weather conditions.

    * **Receive Support Alert Notifications:** Extends the base notification functionality to inform users of updates or issues related to ministry support programs.

Register/Login

* Actor: Farmer/Livestock Breeder

* Description: Allows users to create accounts and log into an account securely to use the features of the system.

Reset Password

* Actor: Farmer/Livestock Breeder

* Description: Allows users to recover or reset their passwords in case of credential issues.

Enter Crop/Livestock Data

* Actor: Farmer/Livestock Breeder

* Description: Allows entering the details of crop and livestock data, which can be tracked and analyzed.

Get Ministry Supports

* Actor: Farmer/Livestock Breeder and Ministry Employee

* Description: Allows users to access information on government-provided agricultural support programs, funding, and assistance offered by the Ministry of Agriculture and Forestry.

Flag Issues in Supports

* Actor: Ministry Employee

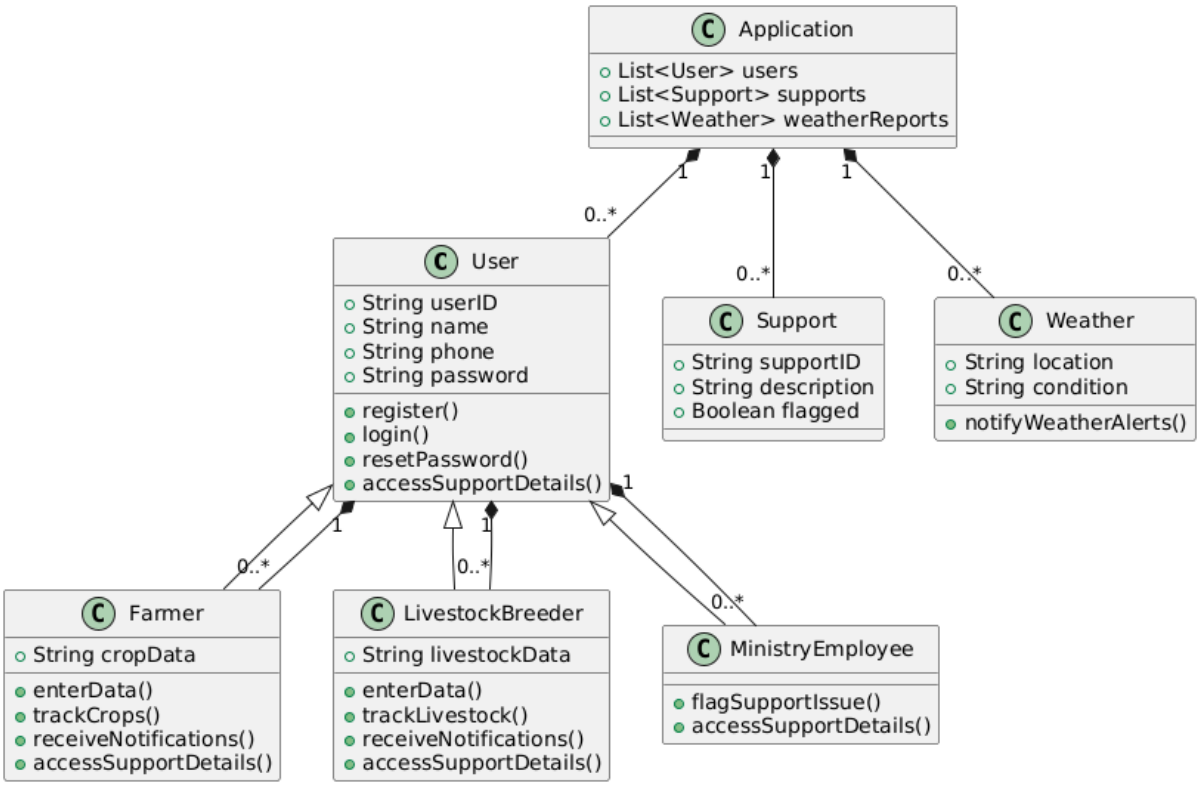
* Description: Makes it possible to identify and flag any discrepancies or issues in support programs for resolution.

Manage Application Settings

* Actor: System Administrator

* Description: Makes it possible for administrator to configure and manage system settings to ensure smooth operation of the ALSApp system.

2.5.3 Object and Class Model



Application Class

This class is the central hub for managing users, support data and weather reports. As seen through the visual, it has associations with the User, Support, and Weather classes.

Attributes:

- List<User> users: It contains a list of all users in the system.
- List<Support> supports: It contains a list of all government support information available in the system.
- List<Weather> weatherReports: It contains a list of weather reports that the system can notify users about.

User Class

This class represents all users interacting with the system. It can be associated with multiple roles such as Farmer and MinistryEmployee classes.

- **Attributes**:

  - String userID: It contains a unique identifier for each user.

- String name: It contains the name of the user.
- String phone: It contains the phone number of the user.
- String password: It contains the user's password for authentication.
- register(): It is a function that allows a new user to create an account.
- login(): It is a function that allows user authentication and grants access to the system.
- resetPassword(): It is a function that allows the user to reset their password.

Support Class

This class stores and manages information related to government support programs.

- **Attributes**:
  - String supportID: It contains a unique identifier for each support entry.
  - String description: It contains details about the support program (e.g., funding or subsidies).
  - Boolean flagged: It indicates whether the support entry has been flagged for issues.

Weather Class

This class represents weather updates and provides alerts to users to help them make informed decisions.

- **Attributes**:
  - String location: It contains the location for which the weather data is provided.
  - String condition: It describes the weather conditions (e.g., "rainy," "sunny").
  - notifyWeatherAlerts(): It sends weather notifications to users based.

Farmer Class

This class represents the farmer's role, providing functionalities tailored to managing crops.

- **Attributes**:
  - String cropData: It contains the data related to crops, such as type, quantity, or health.
  - enterData(): (): It is a function that allows farmers to input data about crops.

- trackCrops(): (): It is a function that allows farmers to monitor the growth and health of crops and animals.
- receiveNotification(): (): It is a function that allows farmers to be able to receive notifications (e.g., weather alerts or support updates).
- accessSupportDetails(): It views detailed information about government support.

LivestockBreeder Class

This class represents the livestock breeder's role, providing functionalities tailored to managing livestock.

- **Attributes**:
  - String livestockData: It contains the data related to livestock, such as health records or productivity
    enterData(): (): It is a function that allows farmers to input data about livestock.
  - TrackLivestock(): (): It is a function that allows farmers to monitor the growth and health of animals.
  - receiveNotification(): (): It is a function that allows livestock breeders to be able to receive notifications (e.g., weather alerts or support updates).
  - accessSupportDetails(): It views detailed information about government support.

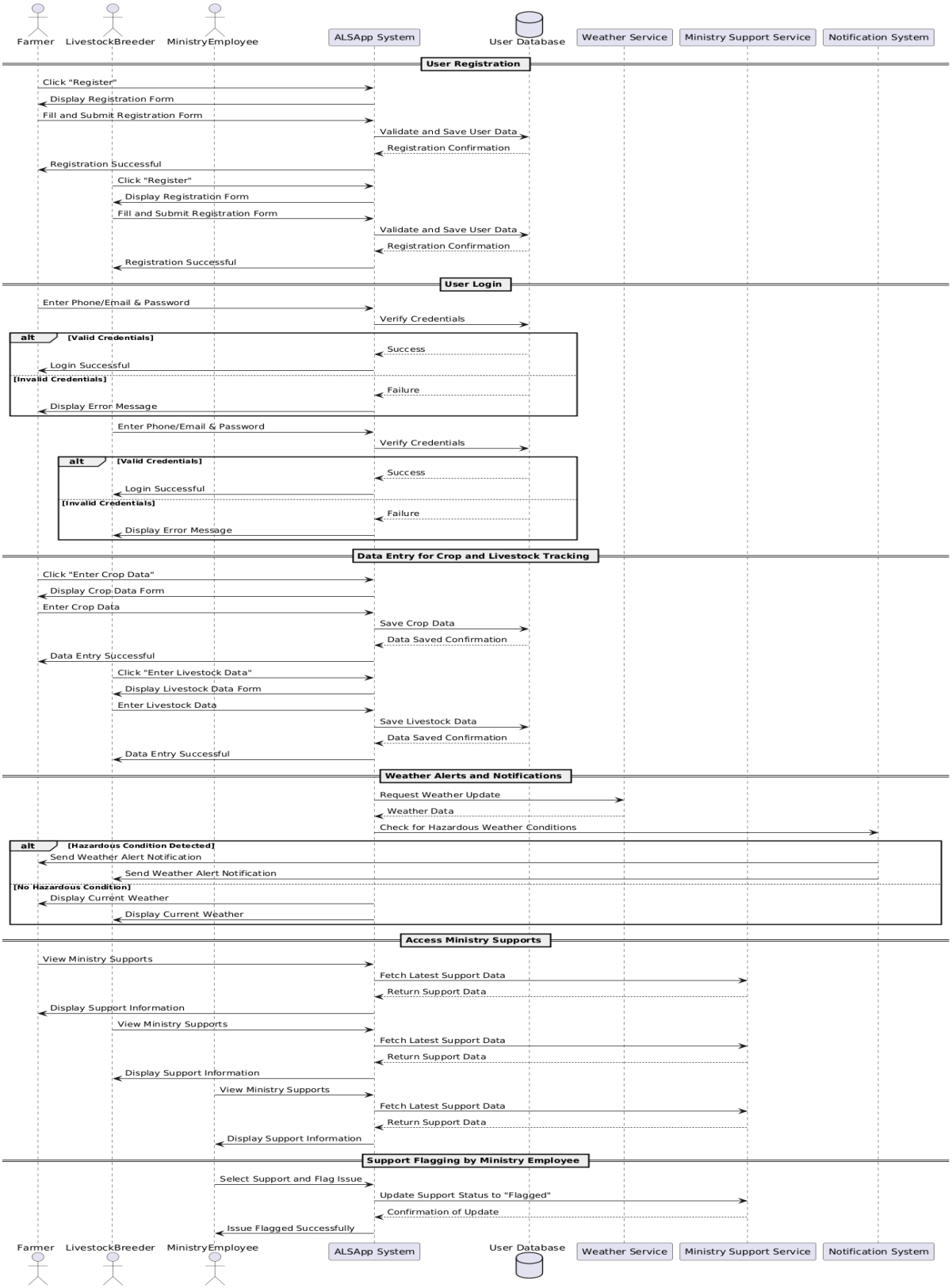
MinistryEmployee Class

This class represents the Ministry of Agriculture and Forestry employees who monitor and manage support processes.

- **Attributes**:
  - flagSupportIssue(): It marks support records as requiring attention or correction.
  - accessSupportDetails(): It views detailed information about government support.

2.5.4 Dynamic Models

a) Sequence Diagram:



This sequence diagram is structured to illustrate how the different entities, systems, and services will be able to interact with its workflow. The detailed information for the diagram is given below:

Key Actors and Participants

Actors:
Farmer
Livestock Breeder
Ministry Employee: Government or support official.

System Components:

-ALSApp System (App): The primary system handling user requests.
-User Database (DB): Stores user credentials, crop, and livestock data.
-Weather Service: Provides real-time weather updates.
-Ministry Support Service: Manages ministry-related support information.
-Notification System: It will send notifications and alerts to the users.

Data Flow Diagram Description

1. User Registration

Farmer Registration:
-The Farmer registers to the system by clicking "Register".
-The App presents the registration form.
-The Farmer fills the required form and submits it to the system.
-App checks and stores user data in the User Database.
-The system confirms to the Farmer that he/she has successfully registered.

Livestock Breeder Registration:
The use case is similar to Farmer registration. However, the actor in this use case is the Livestock Breeder.

2. User Login

Farmer Login:
-Farmer enters their phone/email and password.
-App verifies the credential with the User Database:
-Valid Credentials: Farmers Logged In.
-Invalid Credentials: Error message will appear.

Login for Livestock Breeder:
The entire process is similar to Farmer Login, but this time, the actor would be the Livestock Breeder.

3. Crop and Livestock Tracking Data Entry

Farmer's Entry of Crop Data:
-Farmer selects entering crop cata.
-App will direct the user to crop data entry form.
-Farmer enters crop data and gets stored in the User Database.
-The system notifies the Farmer about the successful entry of data.

Livestock Data Entry:
-The Livestock Breeder selects entering Livestock Data.
-The App displays a livestock data entry form.
-The Livestock Breeder inputs the livestock data and saves into the User Database.
-The system notifies the Breeder that the data has successfully been captured.

4. Weather Alerts and Notifications

-The App initiates a request for the current weather updates from the Weather Service.
-The Weather Service supplies the App with real-time weather data.
-The App checks whether hazardous weather conditions exist in:
Hazardous Condition Detected:
-The Notification System sends weather alerts to both the Farmer and the Livestock Breeder.
No Hazardous Condition:
The App displays current weather conditions to both the Farmer and the Livestock Breeder.

5. Access Ministry Supports
-The farmers and livestock breeders can reach to the supports and display them.
-The Employee of the Ministry views the details of ministry support.
-The App fetches the newest support data from the Ministry Support Service.
The farmers and livestock breeders can be able to display the supports in detail.

6. Flagging Support by Ministry Employee

-The Ministry Employee selects a particular support issue and flags it in the system.
-The App sends a request to update the support status of the Ministry Support Service to "Flagged.".

b) Activity Diagram:

For better understanding the events that occurs in activity diagram is expained in details below:

1. Starting Point

Process begins when the User starts ALSApp.

2. User Registration and Login

The system checks if the users information is existing with the registered user:
-Yes:
The user continues to Log in with credentials.
If the credentials are valid, the user will be able to reach to the Main Dashboard.
If the credentials are invalid, an error message is displayed.
-No:
User is forwarded to the process for Register a new account.
User details are stored in the database and it forwards the user back to the login screen.

3. Selection of Action
Upon Login, the user will be able to utilize three possible actions:
Track Crops/Livestock
Check Weather
Access Ministry Supports

4. Track Crops/Livestock
User enters product/animal data
System checks if data is validated:
-Yes:
The data is saved to the database.
The system analyzes the data for trends or warnings.
If risks are detected, notifications are sent to the user.
If no risks are found, it proceeds to continue monitoring the data.
-No:
It displays an error message.

5. Check Weather
The system gets the live weather data and shows the current weather conditions of a location.
It checks if there is a Weather Alert:
-Yes: A weather alert notification is sent to the user.

6. Access Ministry Supports
The system retrieves the support details.
It checks if the user is a Ministry Employee:
-Yes:
If it’s needed the user can flag a support issue.
-No:
The user will view the support details.

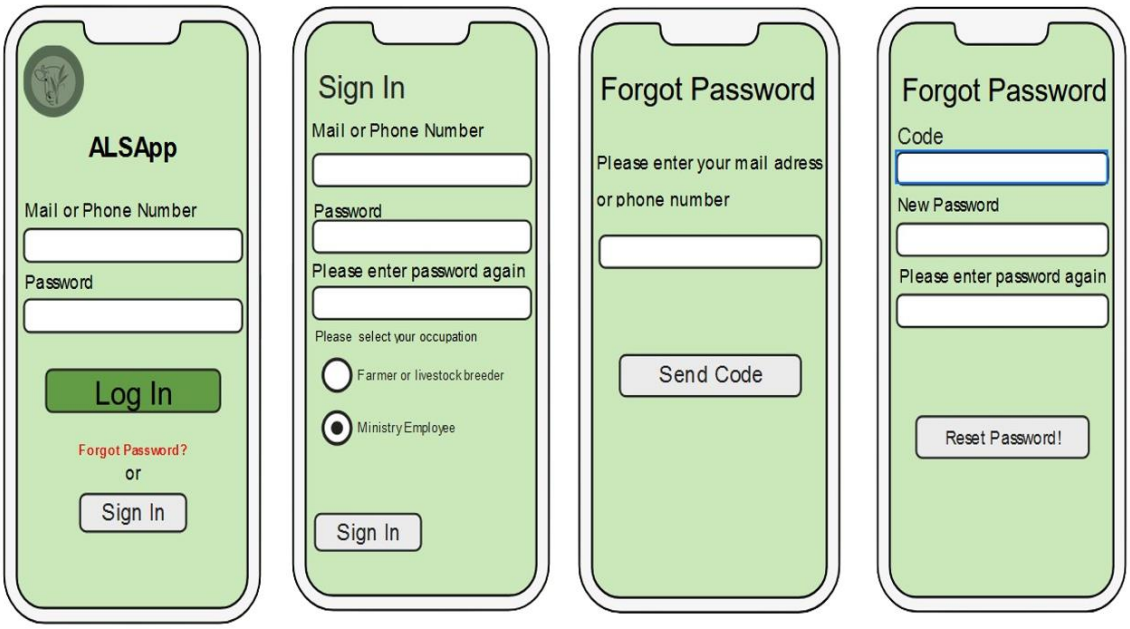
7. Log Out
With any of these selected actions completed, the user will then log out, thus the responsibility of the program will terminate.

2.5.5 User Interface - Navigational Paths and Screen Mock-ups

This section of the report is designed to give the reader an idea about the user interface. The intended use and functions of the different interfaces are explained in detail. In this manner, it is aimed at giving the reader an idea about the function of the application in an easier way.

2.5.5.1 User Registration and Profile Management:

Four different user interfaces are planned for the user to register and log in to the application. These interfaces will be explained in detail below.



1. Application login interface: With this interface, the user will be able to log in to the application with the information stored in the database; if the user is not registered in the application, the user will be directed to the interface created for registration by pressing the 'Sign In' button. Moreover, if the user has forgotten his password, he will be directed to the password renewal interface by pressing the 'Forgot password' button.

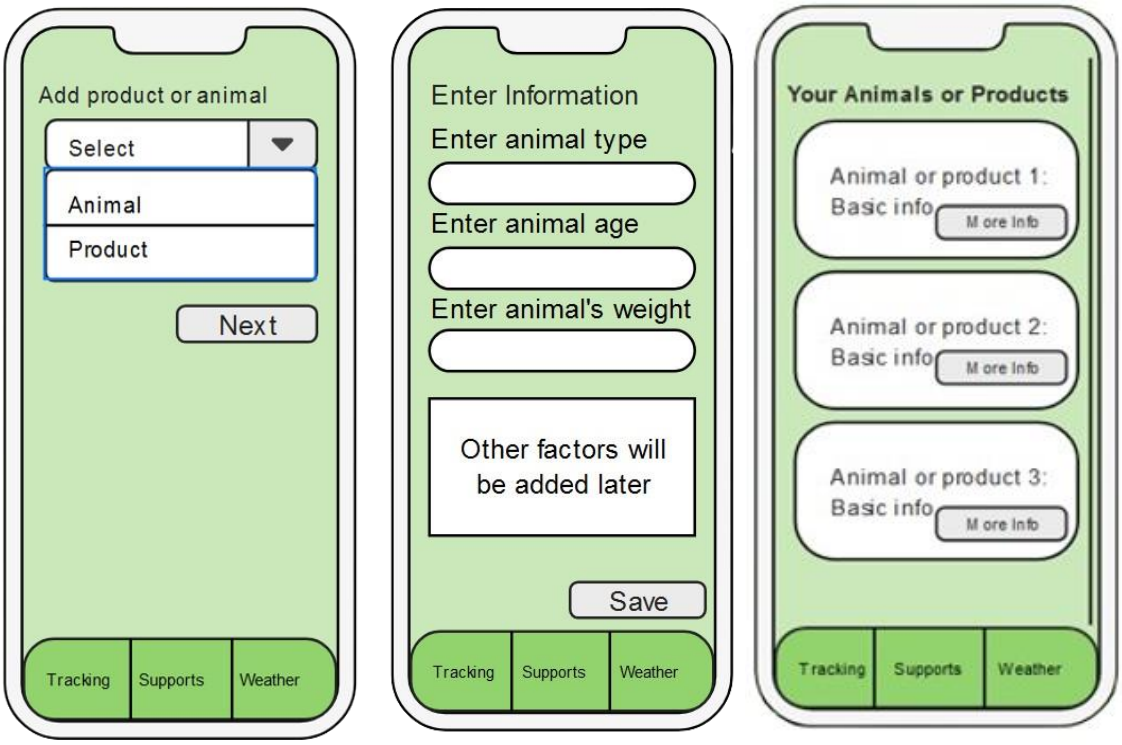
2. Registration interface: After filling in the required personal information in the form, the user will be able to register the application by pressing the 'Sign In' button.

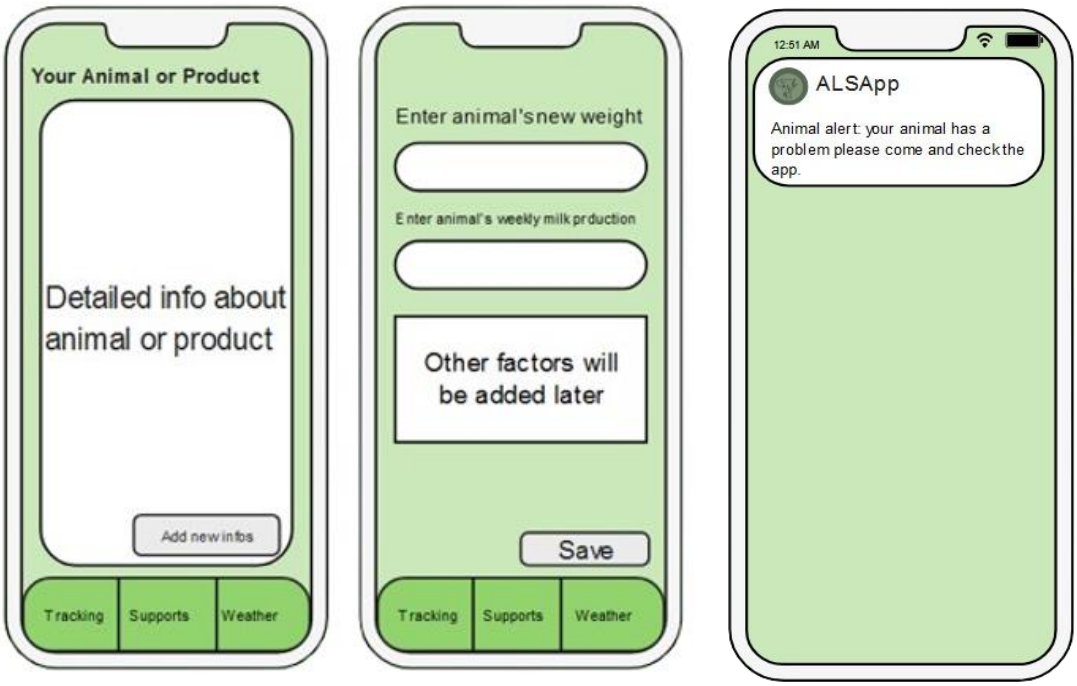
3. Forgot password interface: In this interface, the user will enter their e-mail or telephone address because they have forgotten their password, and they will receive the verification code required when they press the "Send Code" button for two-factor authentication.

4. User authentication interface: After the user fills in the verification code, e-mail address, or phone number sent to them in the form, they will be able to renew their password by clicking the 'Reset Password!' button.

2.5.1.2 Data Entry for Crop and Livestock Tracking:

The interfaces below will enable the farmer to enter data into the application and track the entered products, which are explained in detail in this section.





1. User's selection of animal or product data record: Depending on whether the user is a farmer or an animal farmer, the user selects the option according to whether the data to be entered is animal or product (wheat, legume, vegetable, fruit, etc.) and presses the 'Next' button. Thus, it can proceed to the other interface.

2. Form Interface: The user will be able to save the product information to the database by pressing the 'Save' button with a form as shown in the figure.

3. The interface where animals or products can be viewed: The user can see the animals or products they have previously registered and examine general information about them. If there is a need for more information, the user can click on the 'More Info' button and proceed to the interface with more information.

4. Detailed information interface: Here, the user will be able to receive detailed information about the existing animals or products. If there is a problematic situation, the application will be able to inform the user about it in a thorough information interface. If the user needs to add new information, the user will be directed to the add information interface by pressing the 'Add New Infos' button.

5. Data update: After the user fills in the form containing the necessary data, they will press the "Save" button, and the necessary information will be saved in the database.

6. Notification: If the application encounters any problem with the animal or the product, it will send a notification, as shown in the figure, to inform the user.

2.5.5.3 Weather Alerts and Notifications:
In this section, how the user will be informed about the weather is explained below by adding user interface examples.

1. Weather interface: This user interface will give the weather conditions in the specific location. Moreover, the user will be able to get information about the humidity rate and the possibility of rain. Furthermore, the risk of rain will also be included in the user interface. The application will also evaluate the current season and provide information about the day suitable for planting or harvesting.

2. Notification: The user will be notified with a notification in case of dangerous rainfall or when suitability for planting or harvesting is high.

2.5.5.4 Access to ministry supports:

In this section, user interface examples are given below to help users understand how to access the Ministry's support.

1. Supports interface: Here, the user will be able to see the support of the Ministry, and if they are interested, they will be able to examine the details of these supports by pressing the "More Info" button.

2. Detailed Information Interface: The user will be able to get detailed information about the supports using this interface. If there is any deficiency or problem in the support and the user is a ministry official, they will be able to observe this deficiency more easily.

3. Notification: If a new support is added, the user will be informed about this support through notifications.

3. Glossary

ALSApp: Agriculture and Livestock Support Application, a digital platform for farmers and livestock breeders.

Application: A software program designed to perform specific user tasks, often used on mobile devices or computers.

Livestock: Domesticated animals raised in agriculture for food, fiber, or labor. Examples include cattle, sheep, and poultry.

Real-Time: Data or information that is updated and available immediately as events occur.

Interface: The point of interaction between the user and the application, often designed to be user-friendly and intuitive.

Crop: Cultivated plants grown on a large scale for food, fiber, or other commercial purposes. Examples include wheat, corn, and rice.

Crop Tracking: The process of monitoring and managing crop growth, health, and yield using digital tools.

Sequence: The specific order in which tasks, processes, or events occur within a system or workflow.

Stimulus: An external factor or input that triggers a response in the system, such as a user action or environmental change.

Flagging: The act of marking or identifying data, issues, or conditions that require attention or action, such as potential errors or anomalies.

Compatibility: The application's ability to work seamlessly with different devices, platforms, or systems without conflict.

Scalability: The capacity of the system to handle an increasing amount of work or adapt to growing user demands without compromising performance.

Maintainability: The ease with which the system can be updated, repaired, or enhanced to ensure it remains functional and relevant over time.

Factual Errors: Mistakes or inaccuracies in the data or information provided by the system that can impact its reliability or decision-making.

Yield: The amount of produce harvested from a crop or the output from livestock, often measured per area or animal.

Humidity: A quantity representing the amount of water vapor in the atmosphere or in a gas.

4. References

Sommerville, I. (2007). *Object-oriented software engineering* (3rd ed.). Pearson Education.

Myers, B. (n.d.). Requirements specification. https://www.cs.fsu.edu/~myers/cop3331/notes/requirements.html

rjusama26@gmail.com. (2022, November 27). *Pseudo requirements*. Hudson SteelWorks. https://hudsonsteelworks.com/pseudo-requirements/

Chapter 4 Requirements Elicitation. (n.d.-a). http://www.pa.icar.cnr.it/cossentino/se16-17/ppt/L10_Requirements_Elicitation_ch04lect1-short.pdf

Bandakkanavar, R. (2023, May 8). *Software requirements specification document with example*. Krazytech. https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

AltexSoft. (2023, November 30). *Functional and nonfunctional requirements specification*. https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/