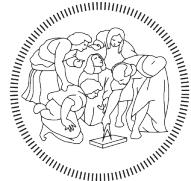


AY 2021/2022



POLITECNICO
MILANO 1863

DD: Design Document

Stefano Brunati: 10623921

Edoardo Cappelletti: 10622565

Gabriele Curti: 10624502

Professor

Elisabetta Di Nitto

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.2.1	Phenomena	4
1.2.2	Goals	4
1.3	Definitions, Acronyms, Abbreviations	5
1.3.1	Definitions	5
1.4	Revision History	5
1.5	Reference Documents	5
1.6	Document Structure	6
2	Architectural Design	7
2.1	Overview	7
2.2	Component view	9
2.3	Deployment view	10
2.4	Runtime view	11
2.4.1	User Sign Up	11
2.4.2	User Login	12
2.4.3	Farmer visualize personalized suggestions	13
2.4.4	Farmer inserts production data	14
2.4.5	Farmer reports a problem	15
2.4.6	Farmer requests for help and suggestions	16
2.4.7	Farmer writes a new post on a forum	17
2.4.8	Farmer subscribes to a forum's topic	18
2.4.9	Agronomist visualizes a daily plan	19
2.4.10	Agronomist inserts a daily plan	20
2.4.11	Agronomist closes a daily plan	21
2.4.12	Policy Maker visualizes farmers' production data	22
2.4.13	Policy Maker asks for best practices	23
2.5	Component interfaces	24
2.6	Architectural styles and patterns	25
2.6.1	Four-tier system architecture	25
2.6.2	RESTful architecture	25
2.6.3	Model View Controller	25
2.7	Other design decisions	26
2.7.1	Thin Client	26
2.7.2	Stateless components	26
3	User Interface Design	27
3.1	User	27
3.2	Farmer	29
3.3	Agronomist	50
3.4	Policy Maker	64
3.5	Flow of functionalities	70
3.5.1	User Sign Up and Login	70
3.5.2	Farmer visualizes weather forecasts	71
3.5.3	Farmer visualizes personalized suggestions	71
3.5.4	Farmer inserts production data	72
3.5.5	Farmer reports a problem	72
3.5.6	Farmer requests for help and suggestions	73
3.5.7	Farmer responds to a request for help and suggestions	73
3.5.8	Farmer writes a new post on forum	74
3.5.9	Farmer responds on a discussion forum	74
3.5.10	Farmer inserts best practices	74
3.5.11	Farmer subscribes to a topic	75

3.5.12	Farmer unsubscribes from a topic	75
3.5.13	Agronomist answers to a request for help and suggestions	76
3.5.14	Agronomist visualizes daily plan	76
3.5.15	Agronomist inserts daily plan	77
3.5.16	Agronomist updates daily plan	77
3.5.17	Agronomist confirms execution of daily plan, or specify deviations	78
3.5.18	Agronomist visualizes farmers' performance (production data)	78
3.5.19	Policy Maker visualizes farmers' performance (production data)	79
3.5.20	Policy Maker asks for best practices to a farmer	79
3.5.21	Policy Maker reports a bad performing farmer	80
4	Requirements Traceability	81
5	Implementation, Integration and Test Plan	83
5.1	Clarification on component integration	86
5.1.1	Integration of the components of the ApplicationServer that exploits the DBMS	86
5.1.2	Integration with external services	86
5.1.3	Integration of RequestManager and ClientManager	87
5.1.4	Integration of Web Browser module and Mobile Application module with the Web Server	88
6	Effort Spent	89

1 Introduction

1.1 Purpose

The purpose of this document is to provide an exhausting explanation about the “software to be”, focusing in particular on the architecture that will be adopted, the modules of the system and their interfaces.

Furthermore, a runtime view of the core functionalities of the “software to be” is provided, accompanied by some detailed interactions diagrams that show the message exchanging between the components.

Finally, there are mentions about the implementation, testing and integration processes.

1.2 Scope

Our goal will be to design and develop a community-centric system that will support the agricultural community via a data-driven approach, bolstering both production and welfare of the farmer population.

Stakeholders of this project will be of three main categories:

- Farmers in the Telangana region. They will be aided in their work from the data that will be available to them. By accessing weather forecasts and critical information when necessary they will be able to both ease their work and get more in return.
- Agronomist involved in aiding farmers in the Telangana region. They will be aided in the organization of their daily visits and in responding to help requests, permitting more mirated and specific work on needing farmers.
- Policy makers in the Telangana region. By seeing specific performance data they will be able to check the results of the rule they applied, and through a direct connection to the farmers they will be able to quickly publish new advice and rules.

DREAM system will bolster Telangana state against food supply shocks and challenges, thanks to the involvement of multiple stakeholders. This will be translated into greater production in order to face the problem of increasing food demand and climate adversities, but also in a way of helping farmers to achieve a better life outside of poverty.

In order to reach greater resilience to meteorological adverse events farmers have access to short and long term forecasts, and also to some “best practices” identified by those farmers who demonstrated to be resilient. Personalized suggestions carry in the same direction, helping to increase both resistance and production. In addition the system also assists farmers to help each other through discussion forums and to request for support and suggestions among themself and to agronomists.

Telangana state is indeed divided into zones assigned to experts (agronomist) exploiting a daily plan to visit the farms of the assigned area (at least twice a year each farm) considering their needs. DREAM system also helps agronomists to visualize and update their plan and analyze the best performing farmers in their area.

All those data, from production to best practices and meteorological resilience, are collected by policy makers, which use them in order to assign special incentives to worthy farmers and to understand if the steering initiatives carried out by agronomists with the help of good farmers produce significant results.

1.2.1 Phenomena

User login	World Shared
User registration	World Shared
Check username and password	Machine
Visualize weather forecasts	World Shared
Visualize personalized suggestions	World Shared
Insert data about production (and problems)	World Shared
Request for help and suggestions by agronomists and other farmers	World Shared
Get notification for help answers	Machine Shared
Respond to a request for suggestions and help	World Shared
Create discussion forums with other farmers	World Shared
Read a discussion forum	World Shared
Respond in a discussion forum	World Shared
Get notification from forum answers	Machine Shared
Receive requests of best practises	Machine Shared
Send best practises to policy makers	World Shared
Work on the crops	World
Get notification for new blog post	Machine Shared
Read blog post	World Shared
Receive incentive notification	Machine Shared
Choose responsibility area	World Shared
Receive help requests from farmers	Machine Shared
Respond with suggestions to farmers	World Shared
Visualize weather forecast in the area	World Shared
Visualize farmer performance data	World Shared
Visualize daily visit plan	World Shared
Modify daily visit plan (before the confirmation)	World Shared
Confirm the execution of the plan	World Shared
Specify deviations from the plan	World Shared
Visit farmers	World
Visualize farmers performance data	World Shared
Request best practices to the “resilient” farmers	World Shared
Receive best practices	Machine Shared
Publish best practice on a blog	World Shared
Decide and send special incentives	World Shared
Visualize crops performance data	World Shared

Table 1: Phenomena

1.2.2 Goals

- G1: Increase the overall welfare and production of the Telangana region. By facilitating the communication and the collaboration between farmers, policy makers, and agronomists the aim is to increase the wellbeing of farmers inhabiting the Telangana region.
- G2: Aid policy makers in the decisional process. Policy makers can see production data in order to decide the incentives for farmers, or whether the current policies are performing well or should be changed (in order to constantly improve Telangana’s production).
- G3: Aid the farmers in the management of their productions. Farmers will receive personalized suggestions and best practices, and they will also have the possibilities to ask for help to both other farmers (by lending/renting equipment or giving advice) or to agronomists.
- G4: Aid agronomist works to help farmers and check crops production. Creating and modifying a daily plan will help them organize their visits and maximize their help in a well-specified zone of expertise.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Farmer: a person who cultivates crops
- Resilient farmer: a farmer whose production is good despite meteorological adverse events.
- Agronomist: an expert in the science of soil management and crop production.
- Policy maker: a person in charge of formulating policies, related to the food system.
- Production: total crops-output generated. Could be related to a single farmer, a zone or the entire Telangana's state.
- Personalized suggestions: indication directly focused on a specific farmer, such as specific crops to plant or specific fertilizers to use based on location and type of production.
- Welfare: overall well-being of farmers which translates into the reduction of poverty and simplification of work (discussion with other farmers, suggestions, personalized data based on location).
- Best practices: cultivation procedure that has been shown by experience to produce optimal results (not only in terms of achieved final production but also in terms of resilience to adversities) and that should be proposed for widespread adoption.
- Responsibility area: zone of which an agronomist is in charge of, with the purpose of increasing its welfare and production.
- Visit: it refers to the agronomist going to a specific farm of his competence, and is identified by a date, a variable time-slot (deviations may occur) and a reason.
- Notification: alert that a certain event has occurred. Could be an email or an automated message sent to the smartphone when the app is not running.

1.4 Revision History

January 8, 2022: version 1.0 (first release)

1.5 Reference Documents

- Specification document: "RDD Assignment A.Y. 2021-2022"
- Course slides
- UML official specification <https://www.omg.org/spec/UML/>

1.6 Document Structure

- Chapter 1: Introduction. This section provides an overall description of the system scope and purpose, together with some information about this document.
- Chapter 2: Architectural Design. This section is addressed to the developer team and offers a more detailed description of the architecture of the system. The first part describes the chosen paradigm and the overall split of the system into several layers. Furthermore, a high-level description of the system is provided, together with a presentation of the modules composing its nodes.
- Chapter 3: User Interface Design. This section is useful for graphical designers of the S2B and contains several mockups of the application, together with some charts useful to understand the correct flow of execution of it.
- Chapter 4: Requirements Traceability. This section acts as a bridge between the RASD and DD document, providing a complete mapping of the requirements and goals described in the RASD to the logical modules presented in this document.
- Section 5: Implementation, Integration and Test Plan. The last section is addressed to the developer team and describes the procedures followed for implementing, testing and integrating the components of our S2B. There will be a detailed description of the core functionalities of it, together with a complete report about how to implement and test them.

2 Architectural Design

2.1 Overview

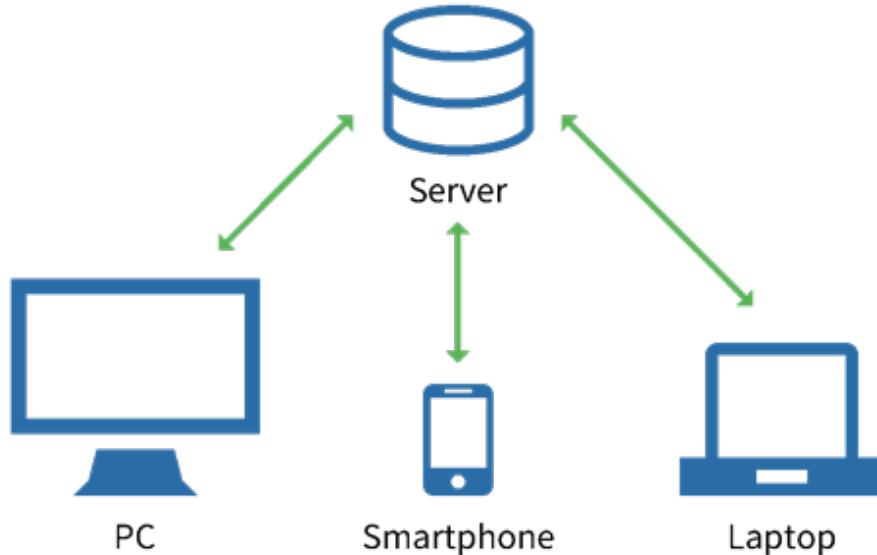


Figure 1: Client-Server architecture

The system is a distributed application which follows the well known client-server paradigm. It implements the thin-client technique, so as to facilitate supporting different client platforms.

There are two main types of clients, the first will be the Web Application, and the second will be the Mobile Application. All the system business logic and data management will be contained and executed in the server.

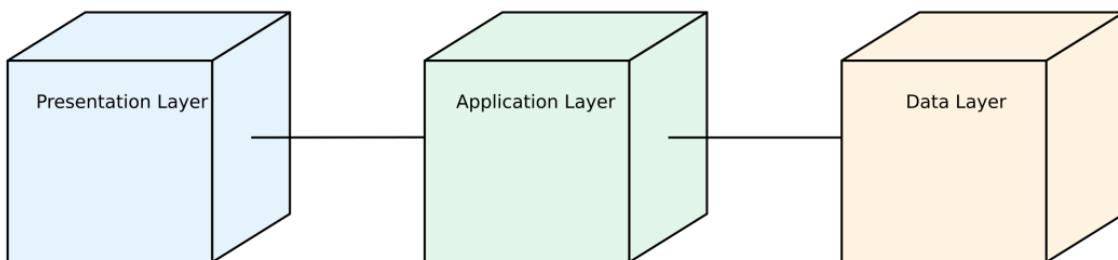


Figure 2: Three-Layers architecture

The system has three layers:

- *Presentation layer*: manages the presentation logic and the user interaction.
- *Application layer*: manages the business logic and functions that the system must provide.
- *Data layer*: manages the storage and retrieval of data.

These layers will be physically separated in the system, and will communicate through known APIs.

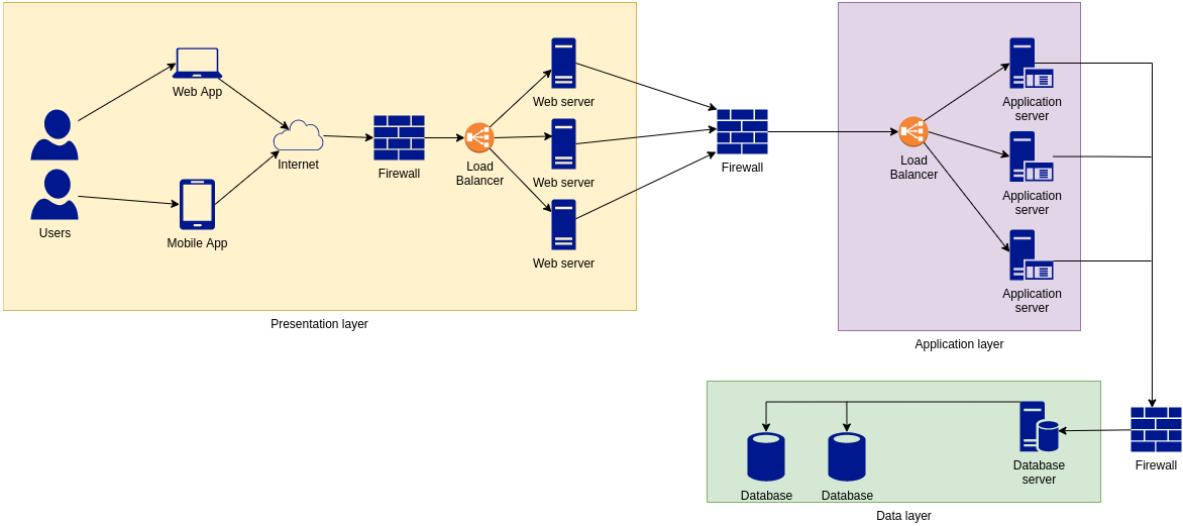


Figure 3: Architecture

The system also has four tiers:

- *Client*: is either a web browser or a smartphone and will render the user interface.
- *Web Server*: manages the incoming connections, prepares the user web pages and communicates requests to the application server.
- *Application Server*: implements the business logic.
- *Data Server*: implements the data storage and retrieval.

The first two tiers are used for the presentation layer, while the 3rd tier is for the application layer and the 4th tier is for the data layer.

The client will communicate with the web servers, which will act as middleware and will interface with the application layer through its APIs. The application layer will then communicate to the data layer using the DBMS APIs.

The application server APIs are RESTful, to better ease the portability of the client code and to guarantee flexibility for scaling. For communicating with the data layer, the ORM technique is used to exploit the object-oriented paradigm to also access the relational database.

Every physical layer will be separated through a firewall and every communication will be encrypted using the HTTPS standard.

The system component will now be described in more depth.

2.2 Component view

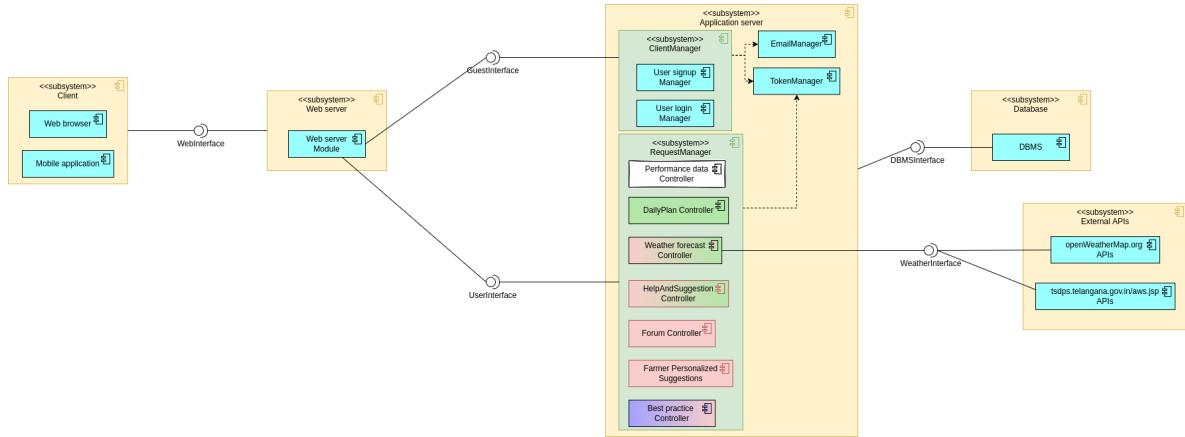


Figure 4: Component View

In the above figure we can see in more detail the components that are part of our system.

- **Web Server**: it has the function of routing requests from the clients to the application server and prepares part of the presentation.
- **Client manager**: it handles all the requests from unauthenticated users, giving them the possibilities of signup or authenticating them. It uses the login interface to manage all the requests and is capable of releasing authentication tokens that will be used for service with the other RESTful APIs. The token will also identify a request for its type of user, if it's a farmer, agronomist or policy maker.
- **Request manager**: it checks a request token and route each request to the correct controller. Each controller has all the functions needed to support its feature. They are color coded to express which user type they must serve, green for agronomists, red for farmers, blue for policy makers and white for all three.
- **Email manager**: it handles the management of email, with built in components for scheduling or sending.
- **Token manager**: it manages, creates, checks and deletes authentication token, used for authenticating the REST requests.
- **DBMS**: it is the relational database controller that manages the storage and retrieval of data, but also the replication and security of such.
- **External APIs**: they represent the APIs of external services, for retrieving weather data.

2.3 Deployment view

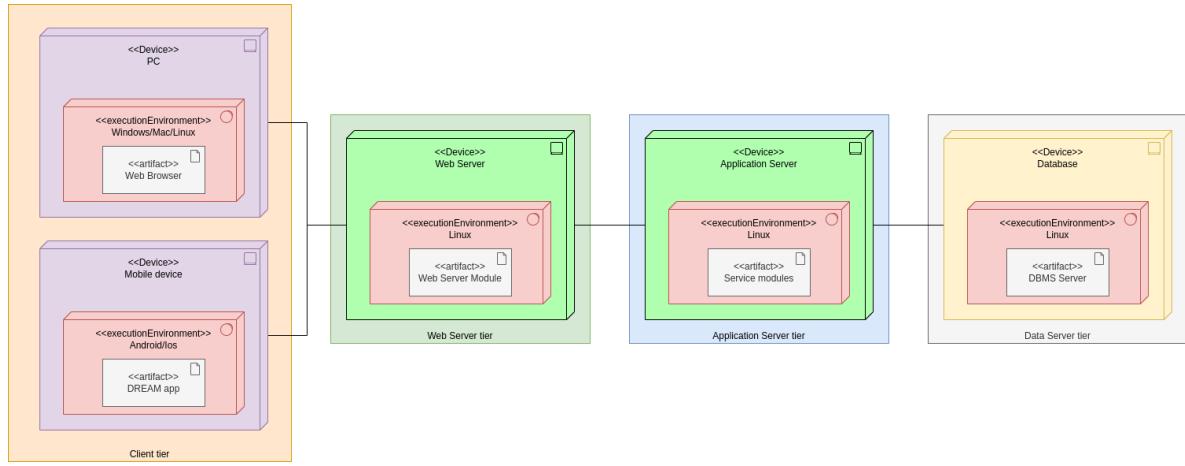


Figure 5: Deployment View

In the above figure is shown the deployment diagram, that shows the needed components for the system to run correctly. Each of these devices has its own operating system where software components run. Copies of these devices can be added to scale out the system.

The tiers are four:

- **Client tier**: these are the client machines, which will run either a browser or the mobile application. To extend the audience the support will be to all major browsers (running on all major os) and mobile operating systems.
- **Web server tier**: this tier includes replicated web servers which receive requests from the clients and routes them to the API of the application server, preparing an HTML page to be rendered on the client, using also client-side scripting and style sheets.
- **Application server tier**: this tier contains replicated application servers which will serve the requests coming from the web servers. It implements all the business logic. To store and retrieve data it will communicate to the Data server tier through the DBMS interface.
- **Data server tier**: this tier includes the machine that will execute the DBMS and the data storage devices. It will execute the requests coming from the application server, storing data securely and with backups options.

2.4 Runtime view

2.4.1 User Sign Up

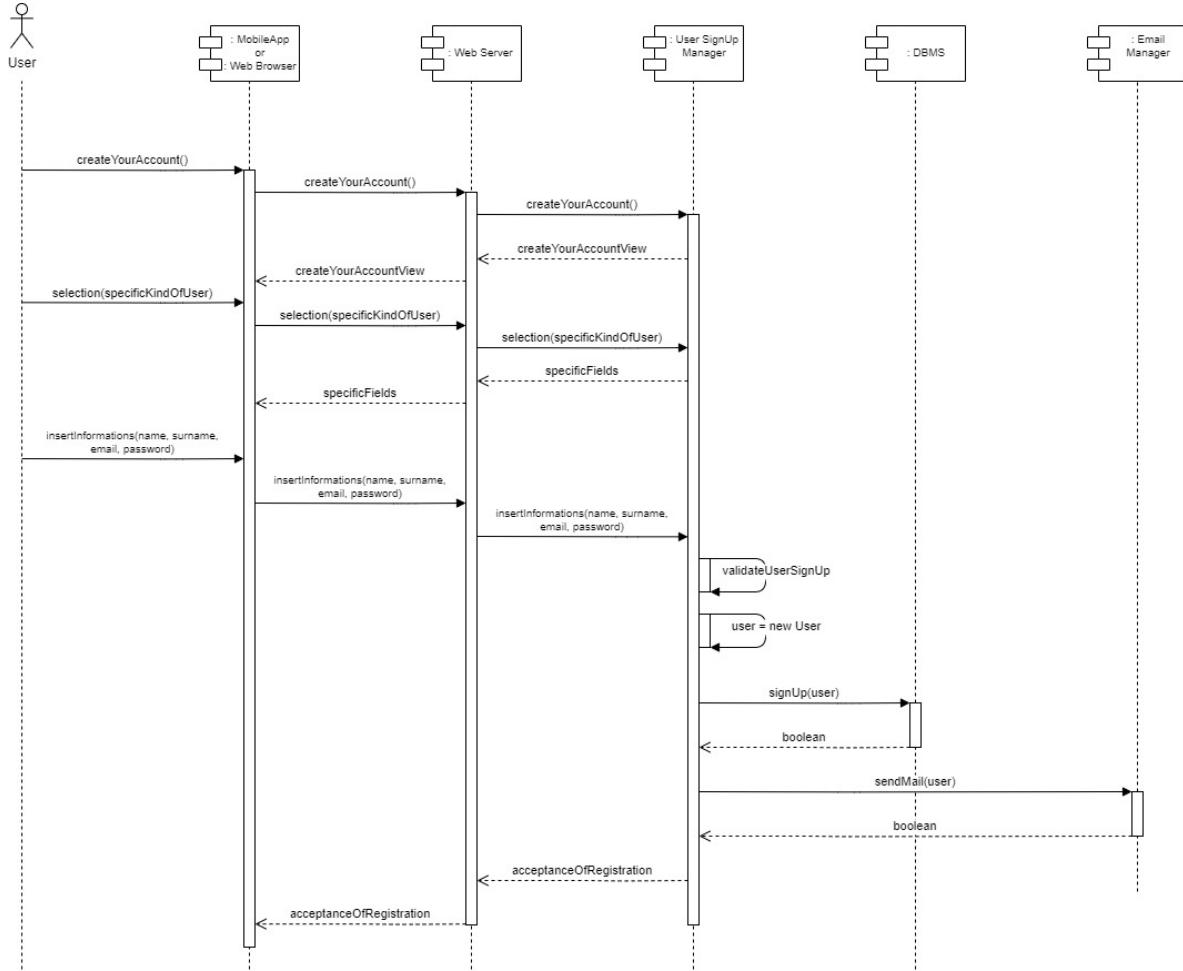


Figure 6: User Sign Up

The diagram above represents the process of signing up a user (Farmer, Agronomist, or Policy Maker).

The three cases are considered together in order to avoid useless repetitions, since the only things that changes are the "specific fields" requested by the system to the user (which are the information inserted lately by the user), according to the selection made by the user itself between the three possible alternatives presented in "createYourAccountView", which are: Farmer, Agronomist, and Policy Maker.

- If the user is a *Farmer*, the specific fields are: name, surname, email, password, farm's name, farm's location.
- If the user is an *Agronomist*, the specific fields are: name, surname, email, password, responsible area.
- If the user is an *Policy Maker*, the specific fields are: name, surname, email, password.

2.4.2 User Login

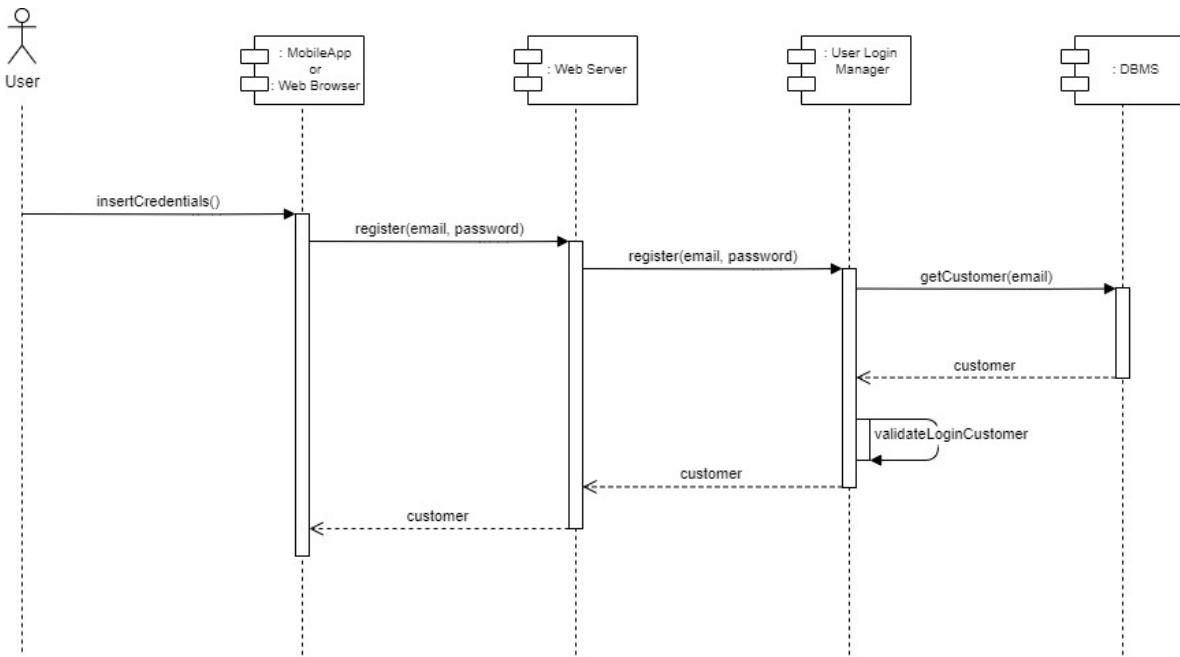


Figure 7: User Login

The diagram above represents the process of login of a user, who can be a Farmer, or an Agronomist, or a Policy Maker.

The user inserts his credentials (email and password), the systems checks if the credentials are correct and then shows the home page according to the user type.

2.4.3 Farmer visualize personalized suggestions

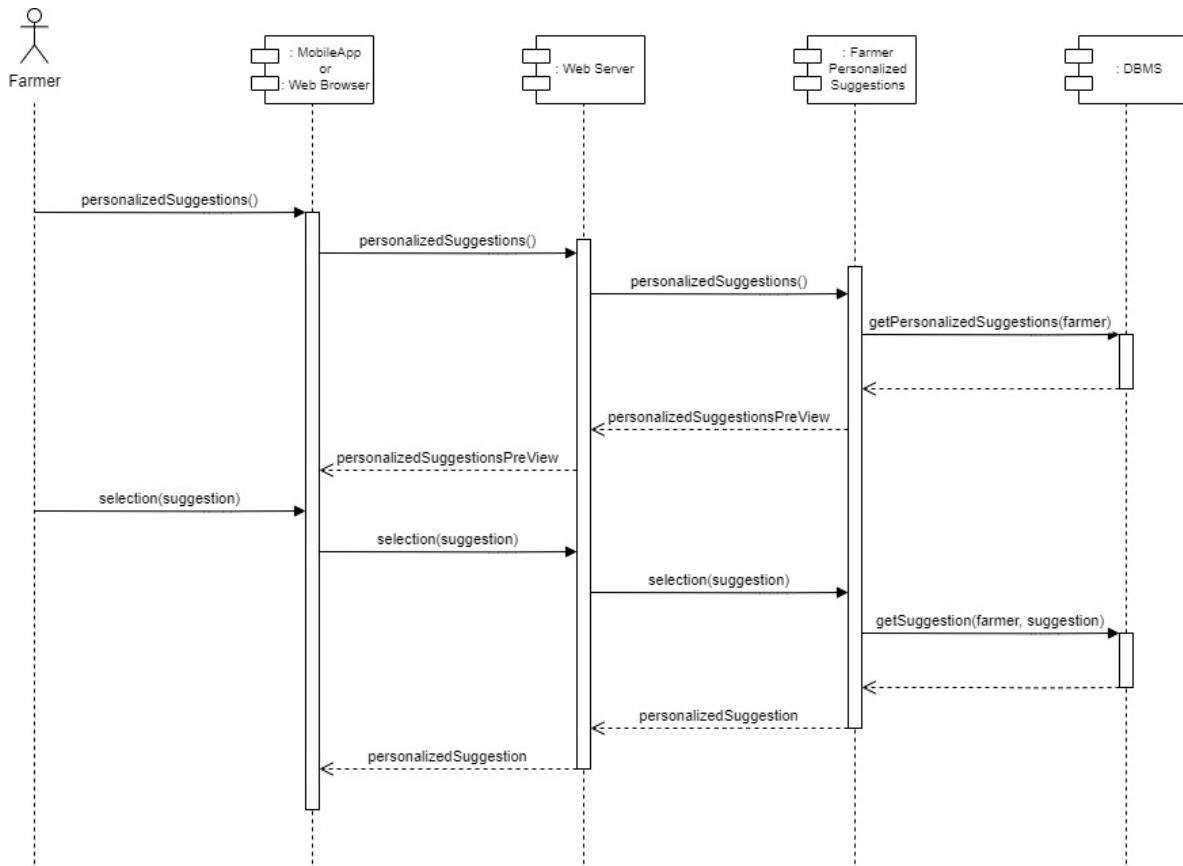


Figure 8: Farmer visualize personalized suggestions

The diagram above represents the process of viewing the personalized suggestions of a farmer, who is already on his home page.

The system displays a preview of all the suggestions, then the farmer selects a specific suggestions through the ones showed, thus the system displays the selected suggestion with all the particulars.

2.4.4 Farmer inserts production data

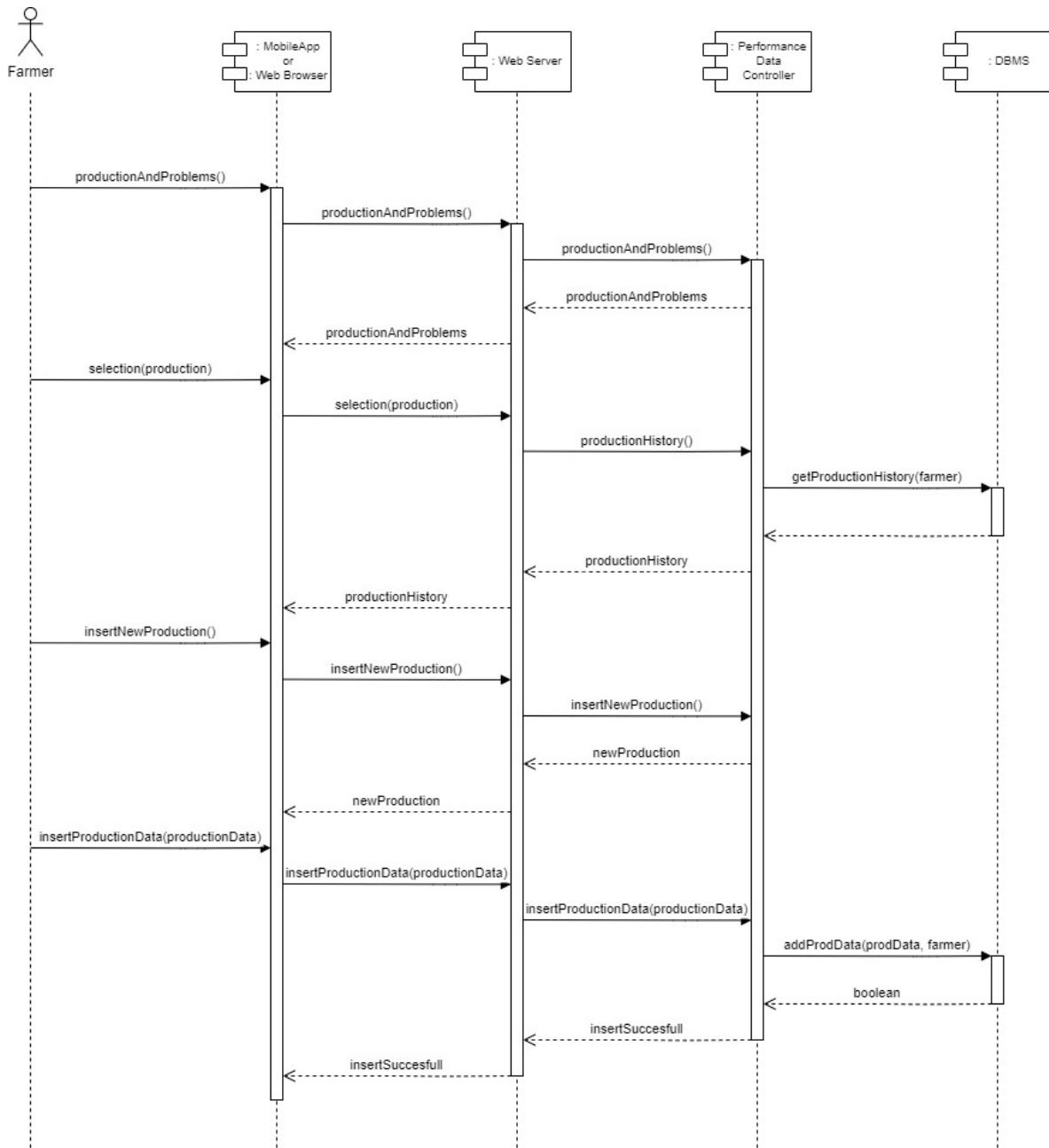


Figure 9: Farmer inserts production data

The diagram above represents the process of inserting into the system the production data by a farmer, who is already on his home page.

The system first asks between inserting production data and reporting a problem, and then, after the farmer selects to insert production data, displays the production data history of the farmer.

The farmer is now able to click on the button "Insert Production Data", and after that the systems shows a list of fields that must be inserted by the farmer to correctly insert the production.

The farmer inserts all the data, so that the system is able to store the production into the DB and, if everything works properly, displays a message saying that the data are stored.

2.4.5 Farmer reports a problem

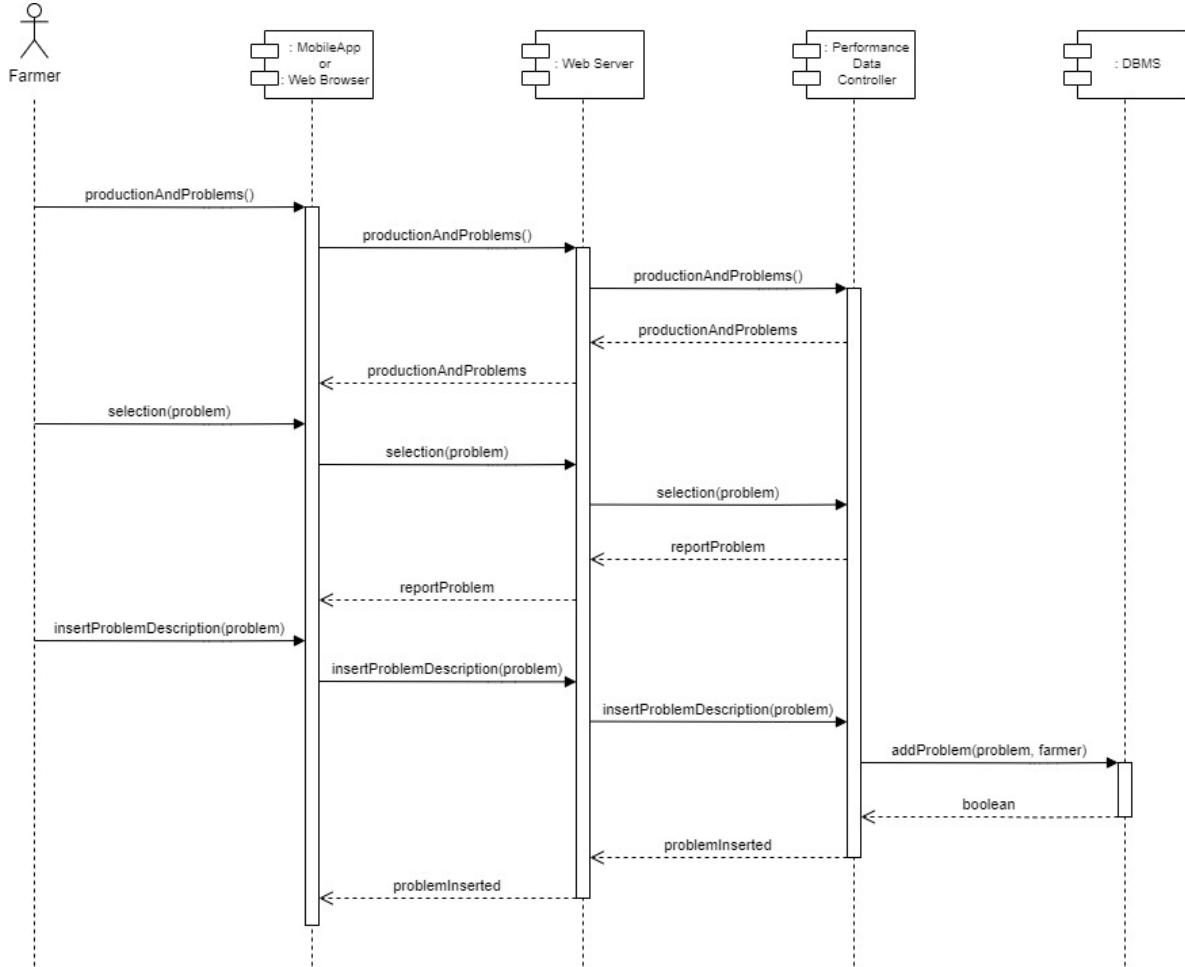


Figure 10: Farmer reports a problem

The diagram above represents the process of reporting a problem by a farmer, who is already on his home page.

The system first asks between inserting production data and reporting a problem, and then, after the farmer selects to report a problem, asks a description of the problem and shows the agronomists responsible of the farmer's area.

The farmer can now insert the description of the problem and selects an agronomist to which request for help.

The system is able to store the problem into the DB and, if everything works properly, displays a message saying that the problem is stored and then the farmer home page.

2.4.6 Farmer requests for help and suggestions

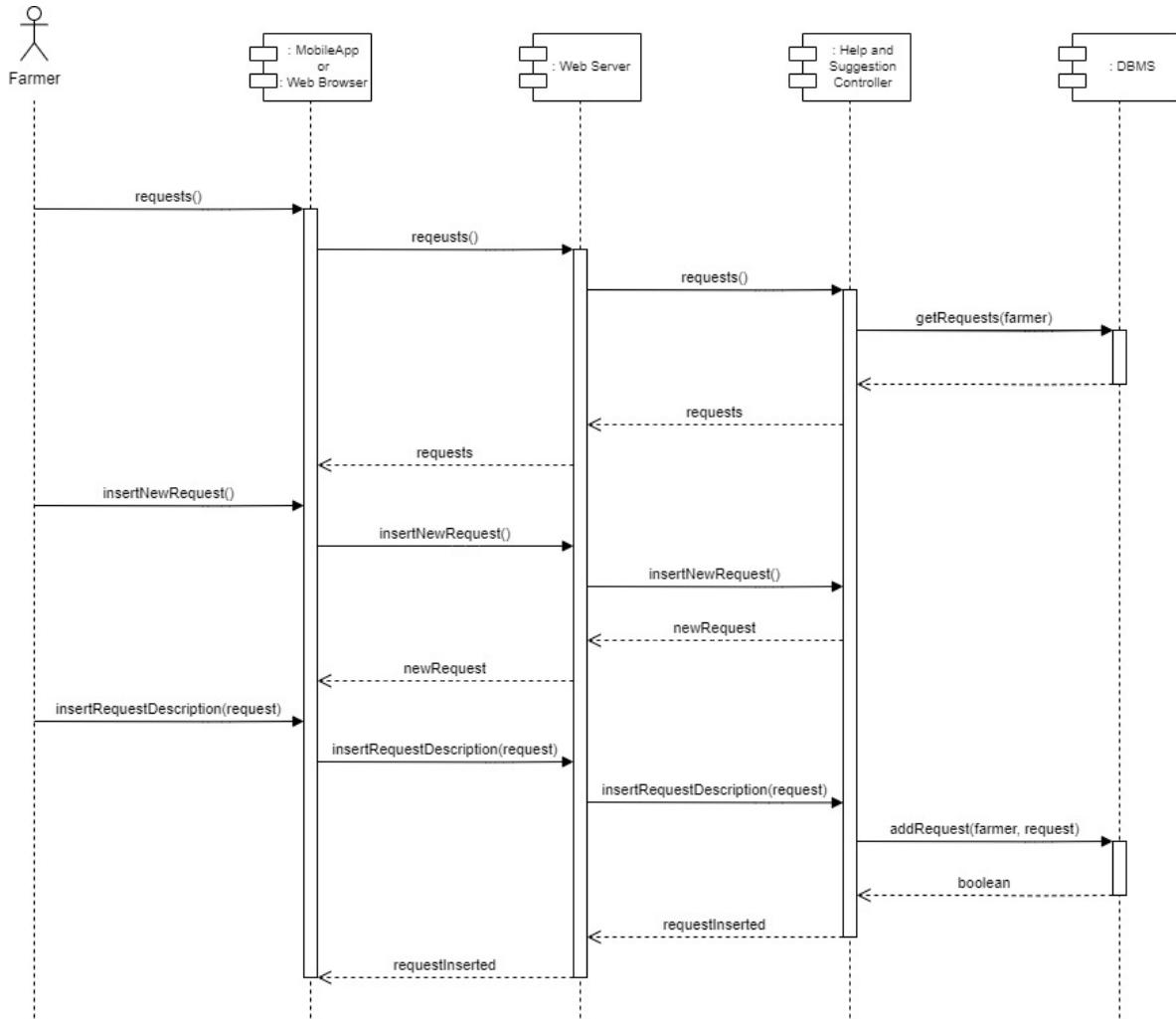


Figure 11: Farmer requests help

The diagram above represents the process of requesting help by a farmer, who is already on his home page.

The system first displays a list of the most recent received, and then, after the farmer selects to make a new request, asks a description of the problem and a preferred receiver (receiver can be either a farmer or an agronomist).

The farmer can now insert the description of the problem and can select a farmer or an agronomist to whom request for help.

The system is able to store the problem into the DB, to contact the selected receiver and, if everything works properly, to display a message saying that the request for help is sent.

2.4.7 Farmer writes a new post on a forum

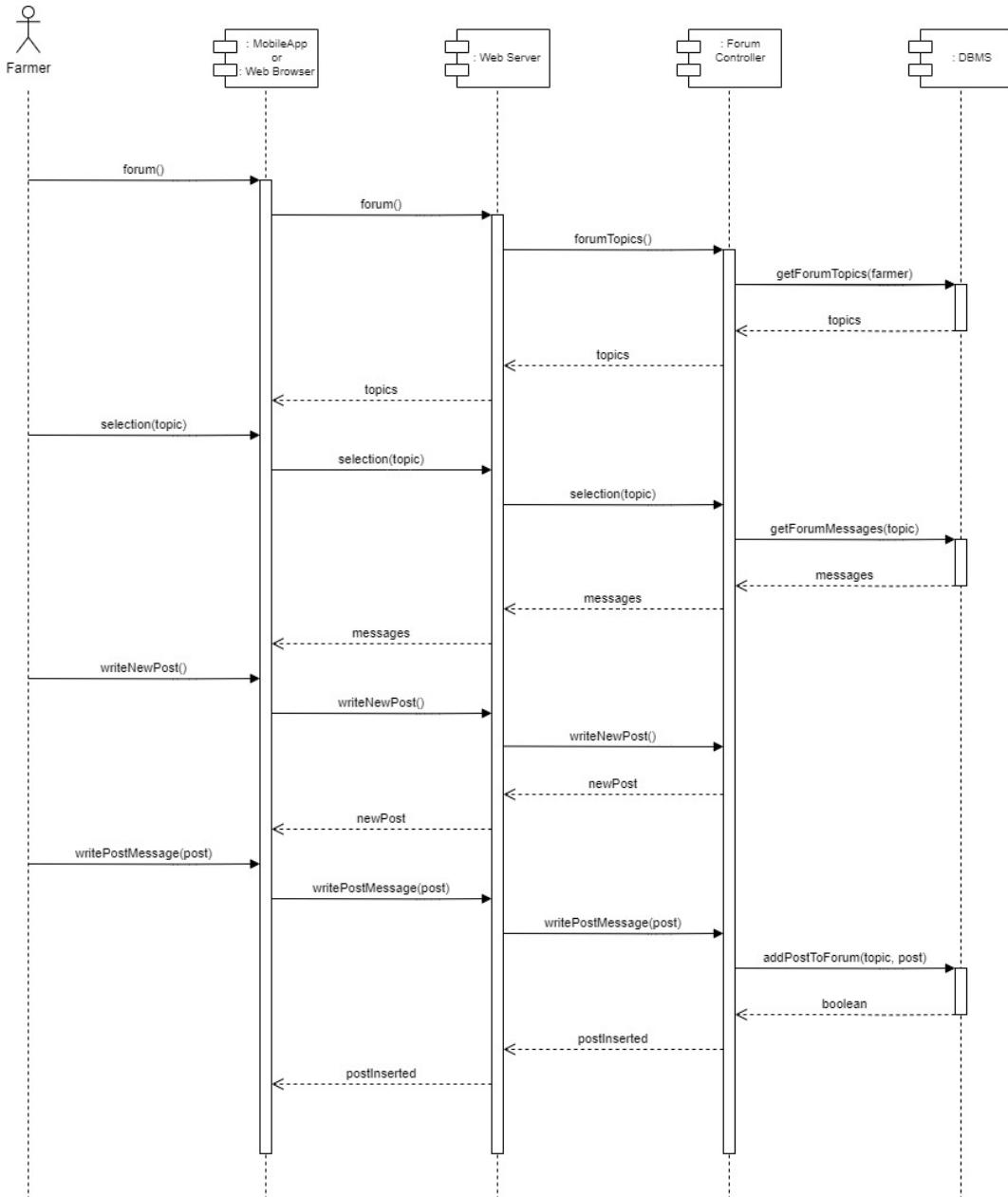


Figure 12: Farmer requests help

The diagram above represents the process of writing a new post on the forum by a farmer, who is already on his home page.

The system first displays a list of all the topics of the forum, and then, after the farmer selects a specific topic, shows the latest messages of the selected topic.

The farmer clicks on the new post button, and then he can write the message; the system store the message into the DB and, if everything works properly, displays a message saying that the post has been created.

2.4.8 Farmer subscribes to a forum's topic

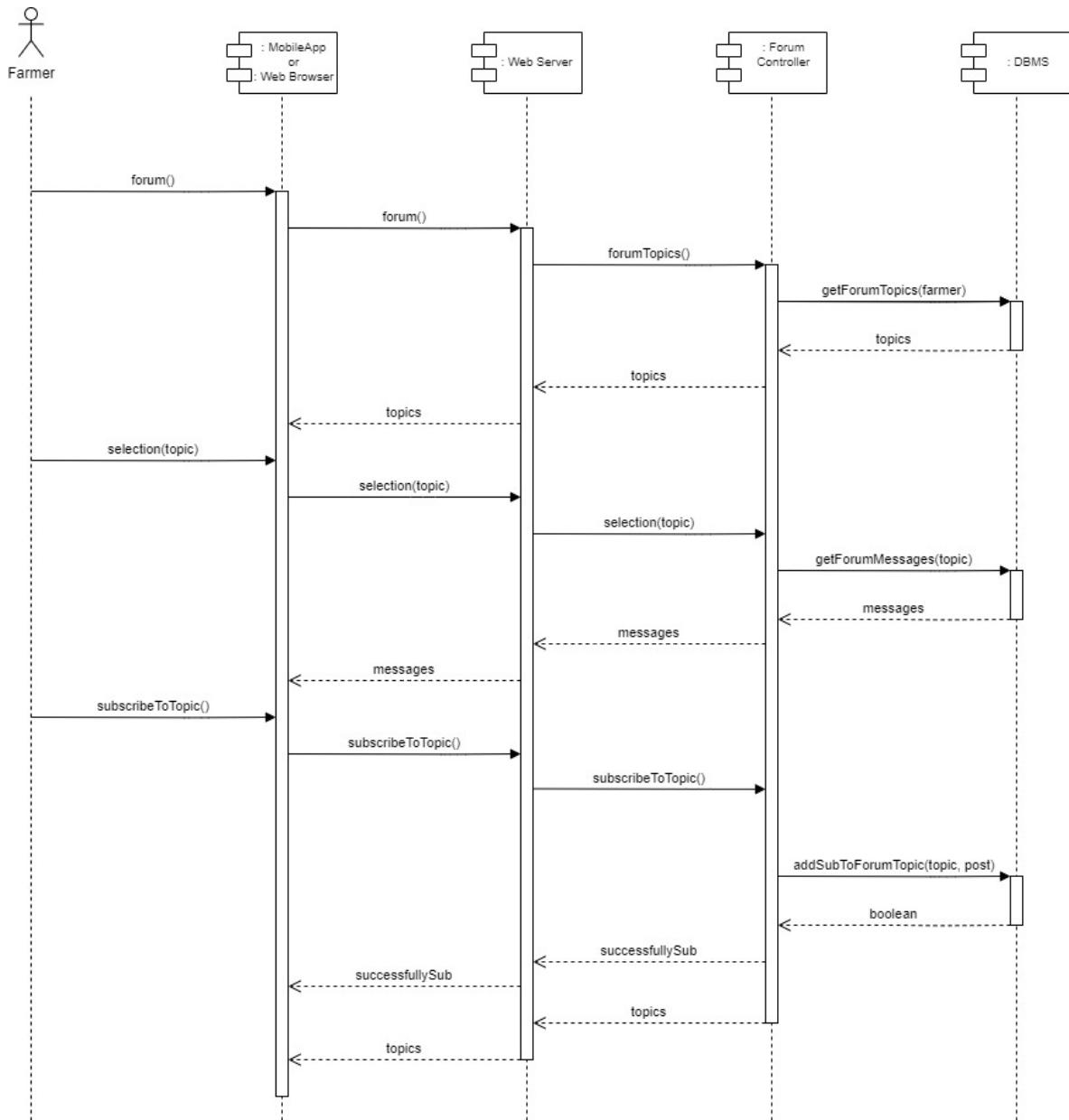


Figure 13: Farmer subscribes to a forum's topic

The diagram above represents the process of subscribing to a topic in the forum by a farmer, who is already on his home page.

The system first displays a list of all the topics of the forum, and then, after the farmer selects a specific topic, shows the latest messages of the selected topic.

The farmer clicks on the subscribe button; the system store the subscription into the DB and, if everything works properly, displays a message saying that the subscription has been successful and then shows the latest messages related to the topic.

2.4.9 Agronomist visualizes a daily plan

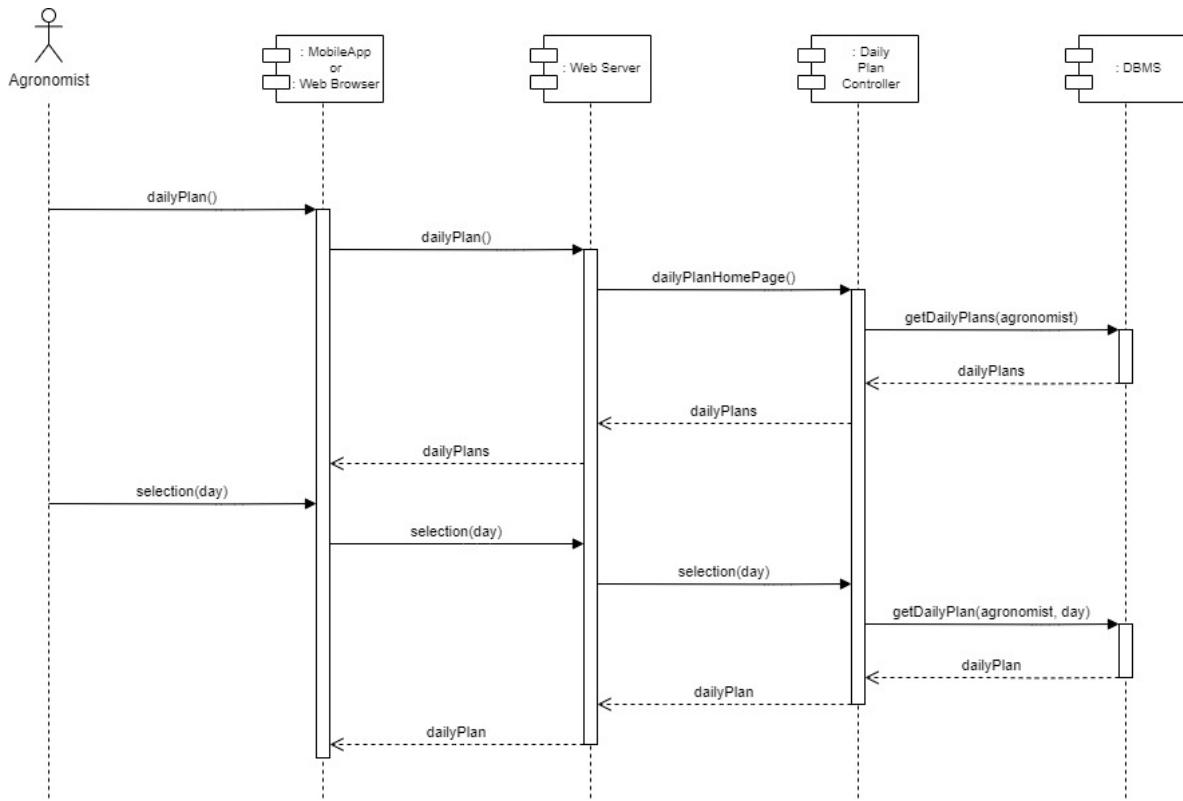


Figure 14: Agronomist visualizes a daily plan

The diagram above represents the process of viewing a daily plan (already inserted) by an agronomist, who is already on his home page.

The system first displays a month calendar with, and then, after the agronomist selects a specific day, shows the daily plan of the selected day.

2.4.10 Agronomist inserts a daily plan

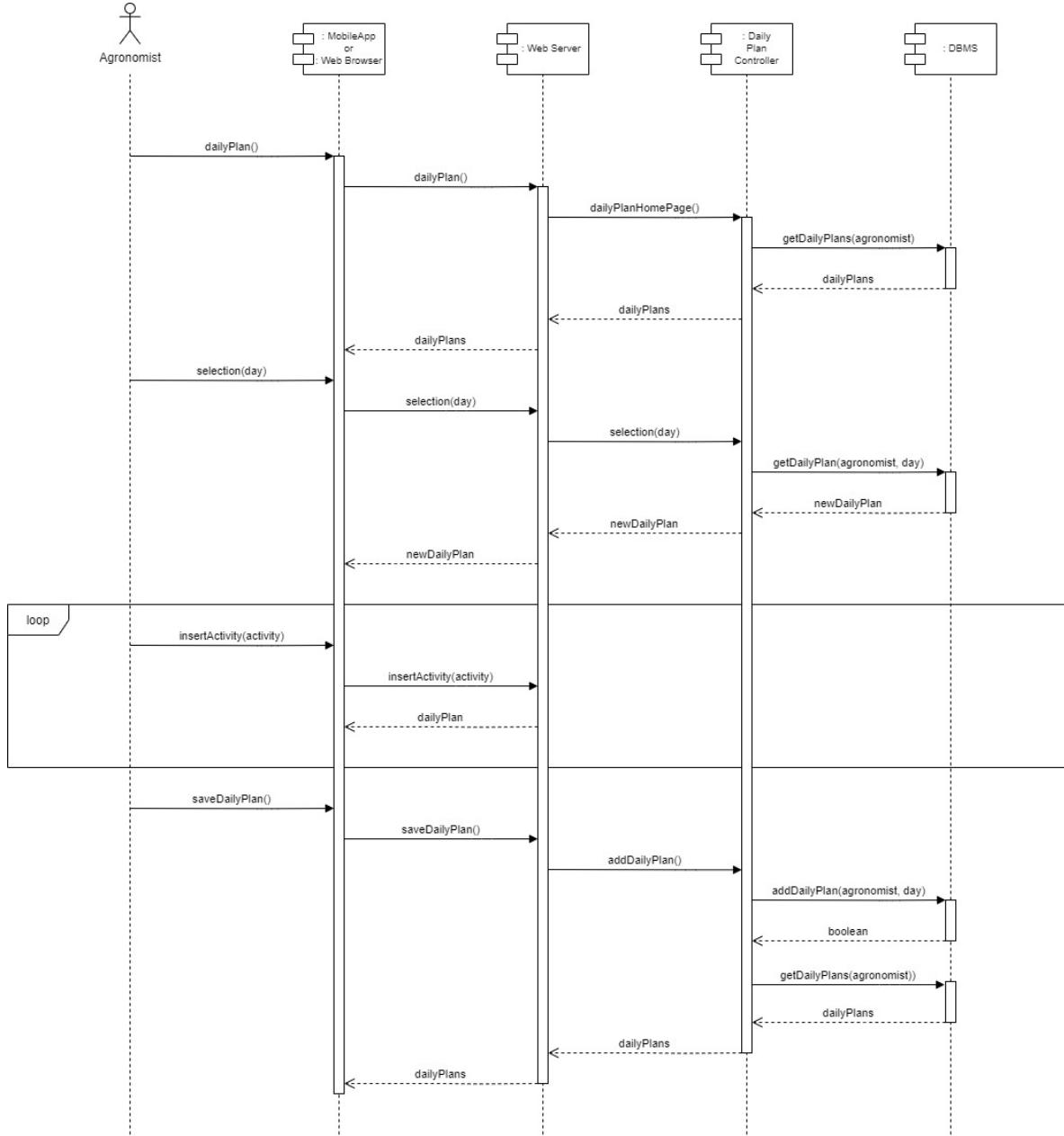


Figure 15: Agronomist visualizes a daily plan

The diagram above represents the process of inserting a daily plan by an agronomist, who is already on his home page.

The system first displays a month calendar, and then, after the agronomist selects a specific day (with no daily plan inserted), allows the agronomist to recursively inserting new activity for the day.

When the agronomist clicks on save button, the system stores the new daily plan into the DB and, if everything works properly, displays a message saying that the daily plan has been successfully inserted and then shows again the calendar view.

2.4.11 Agronomist closes a daily plan

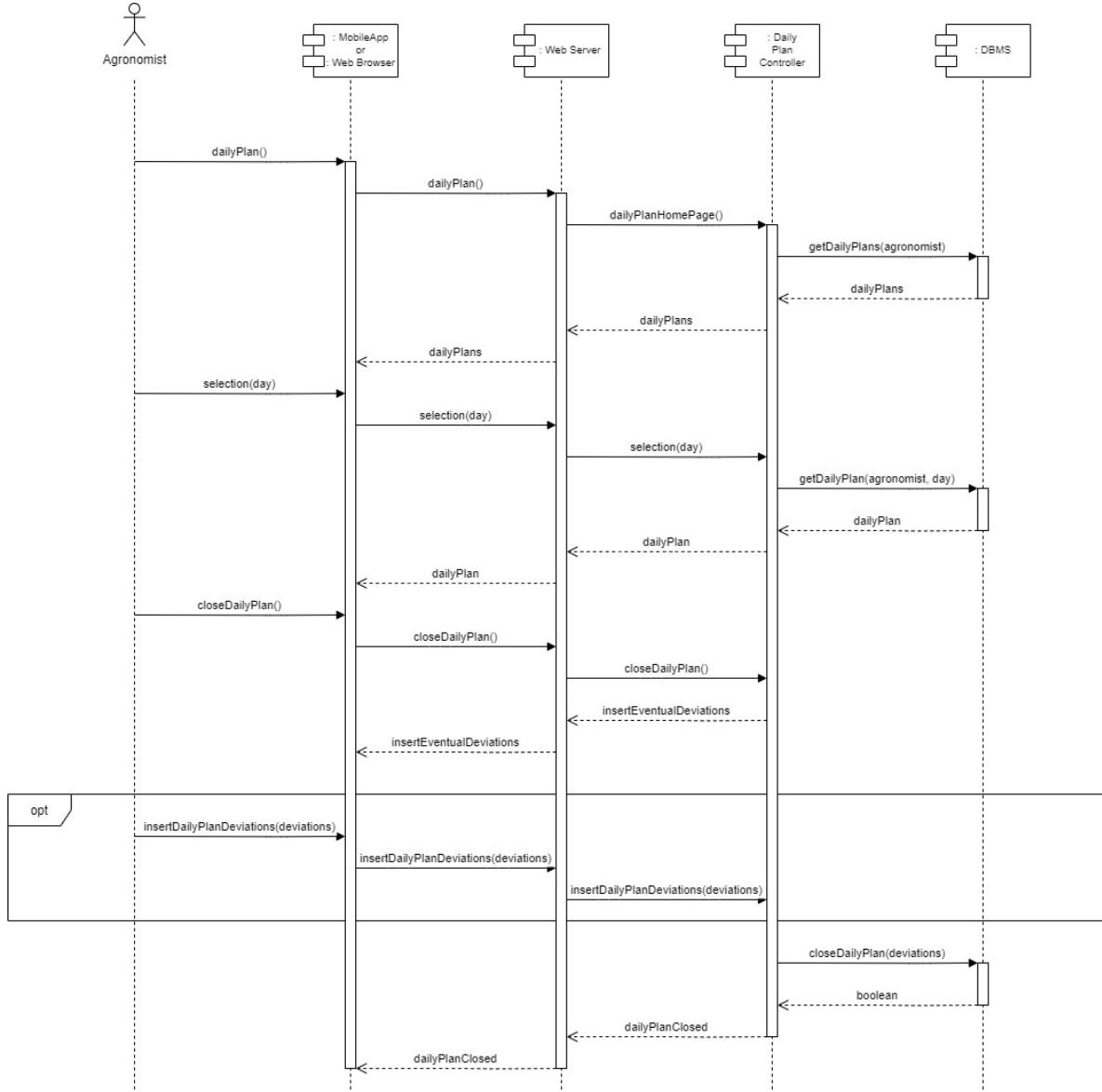


Figure 16: Agronomist confirms the execution of a daily plan (specifying eventually deviations)

The diagram above represents the process of closing a daily plan by an agronomist, who is already on his home page.

The system first displays a month calendar, then, after the agronomist selects a specific day (with a daily plan inserted), shows the daily plan of the selected day.

When the agronomist clicks on confirm and specify deviations button, the system allows to insert optionally some deviations; the agronomist can just click on the confirm button, or can first insert the deviations and then click on the button. The daily plan is closed into the DB and, if everything works properly, displays a message saying that the daily plan has been successfully closed and then shows the agronomist home page.

2.4.12 Policy Maker visualizes farmers' production data

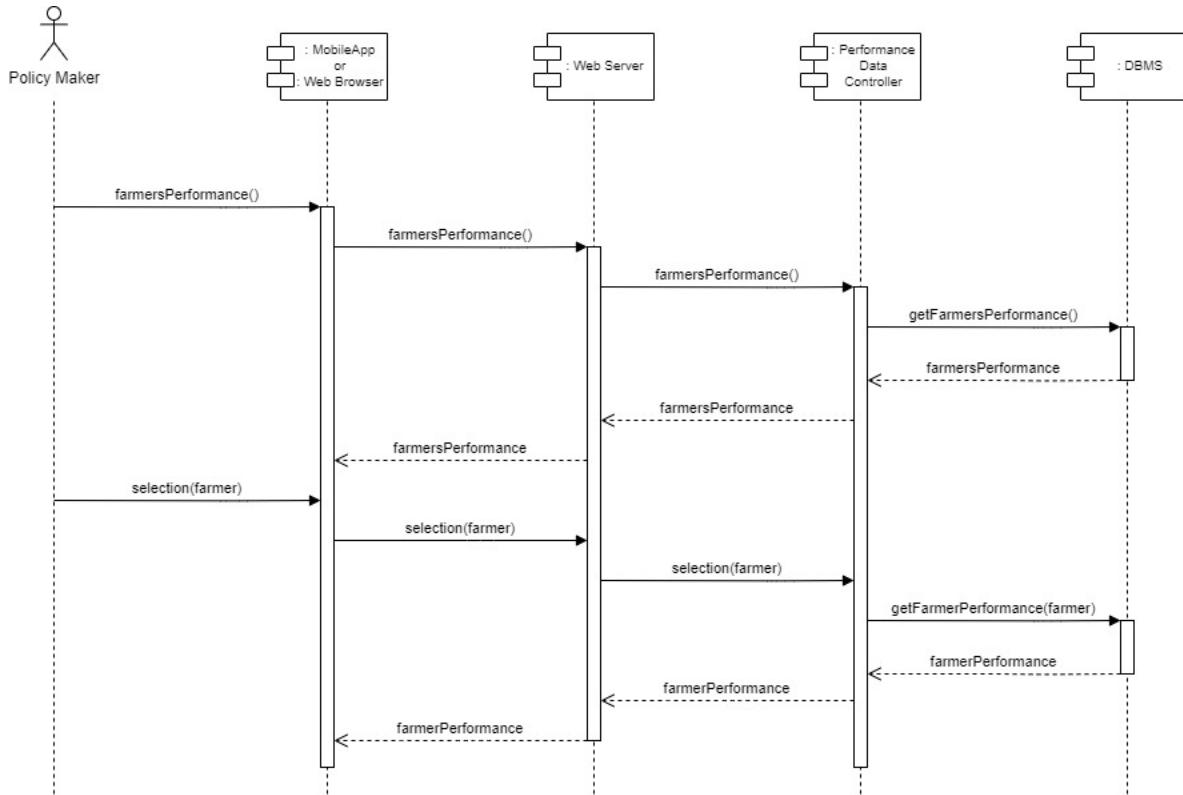


Figure 17: Policy Maker visualizes farmers' production data

The diagram above represents the process of visualizing the production data of farmers by a policy maker, who is already on his home page.

The system first displays a graph with the overall results of the year and a list of farmers ordered by increasing performance, then, after the policy maker clicks on a specific farmer (one on top of the list), shows the production data about the farmer (corn produced, energy, fertilizer, and water used per unit).

2.4.13 Policy Maker asks for best practices

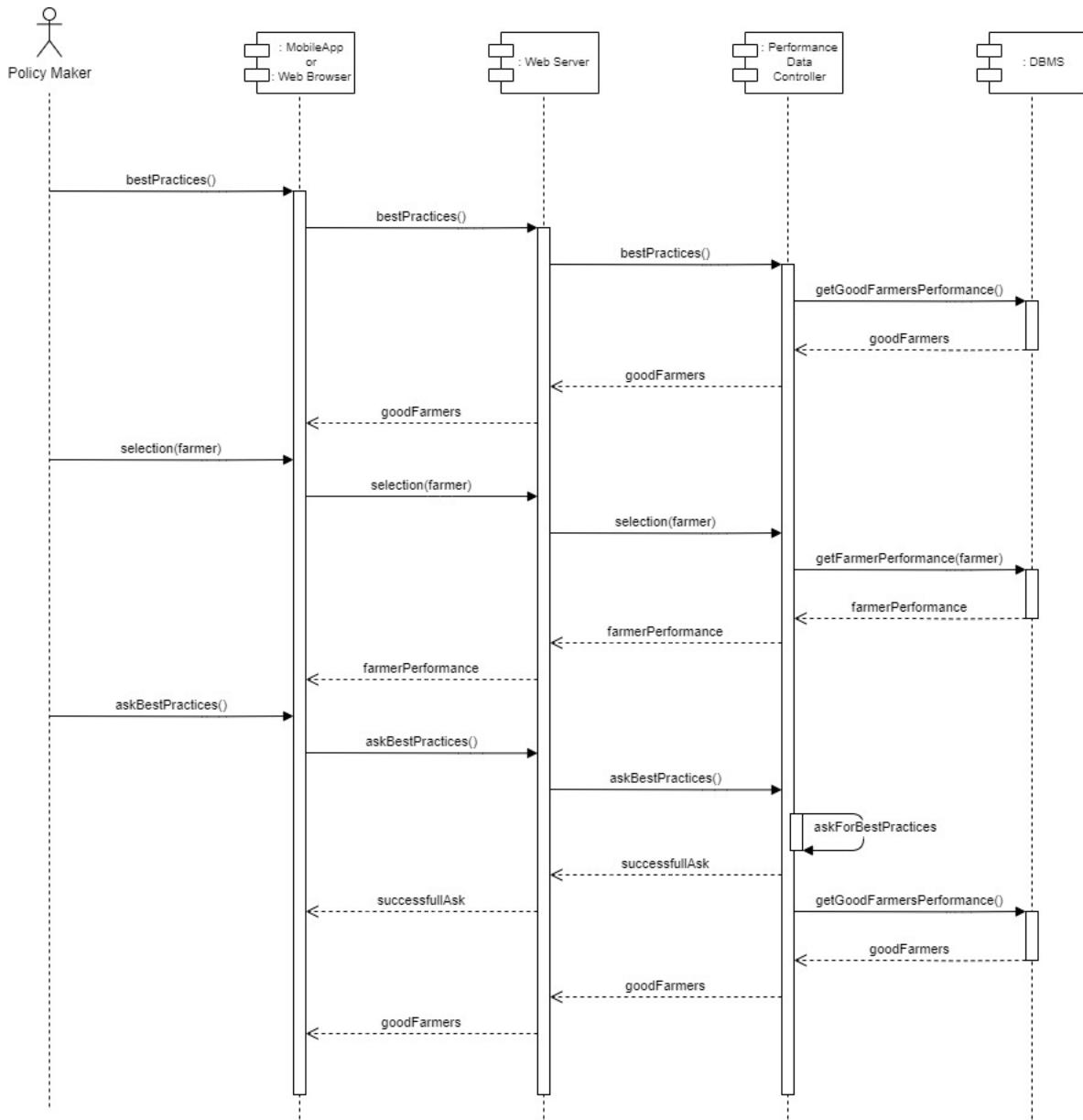


Figure 18: Policy Maker asks for best practice

The diagram above represents the process of visualizing the production data of farmers by a policy maker, who is already on his home page.

The system first displays a graph with the overall results of the year and a list of farmers ordered by increasing performance, then, after the policy maker clicks on a specific farmer (one on top of the list), shows the production data about the farmer.

The policy maker clicks on ask for best practice button, and then the system shows a predefined message to be sent to the selected well performing farmer; when the policy maker clicks on send button, the system send the request to the farmer and shows again the list of farmers ordered by performance.

2.5 Component interfaces

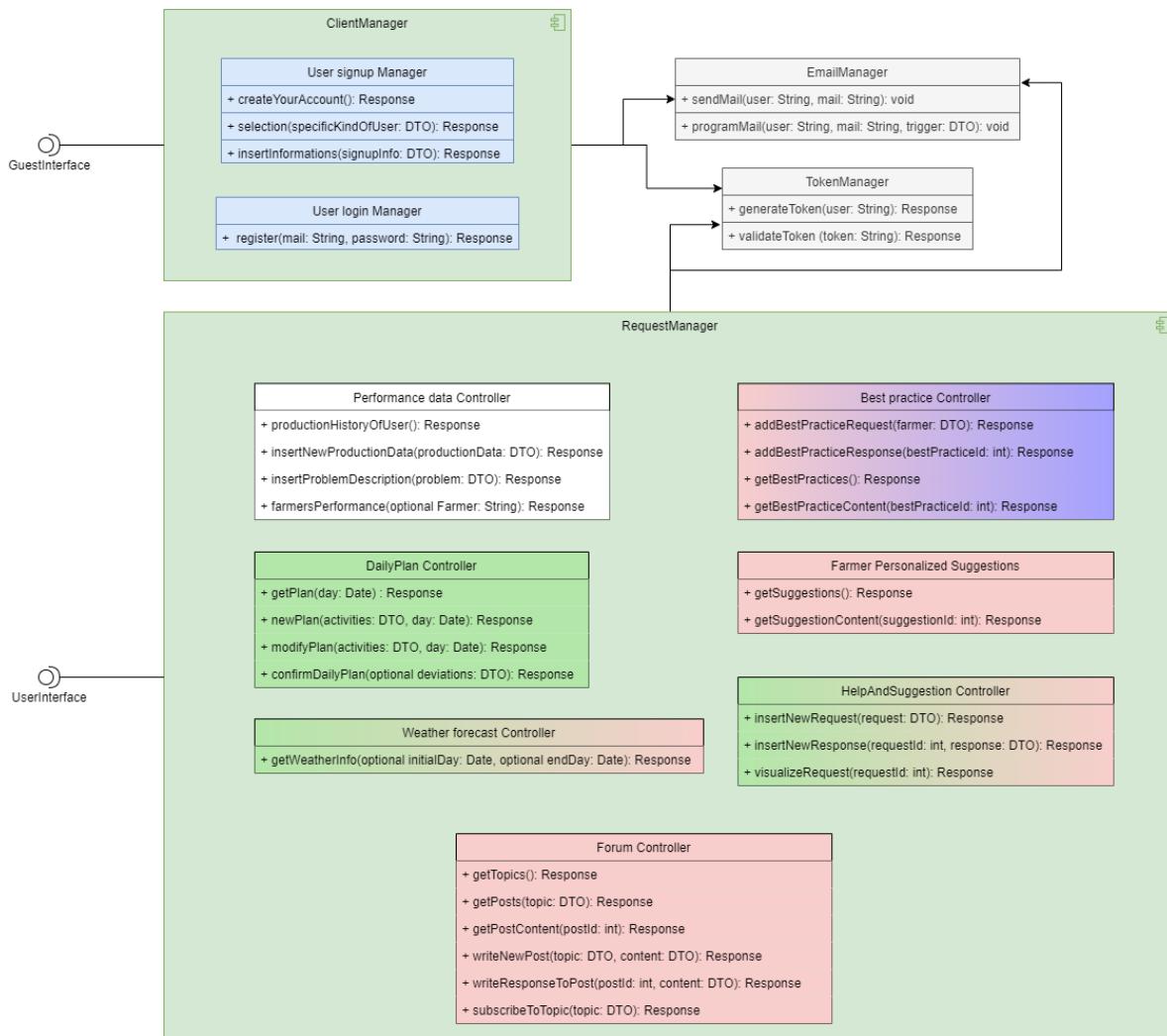


Figure 19: Interfaces Diagram

The interfaces shown in the diagram are the essential one to support the Dream application.

The Guest Interface is used for communication from unauthenticated users, while the User Interface is used for authenticated users communications. Every method through the User Interface will also carry the authentication token for securing the requests and recognising the user.

2.6 Architectural styles and patterns

2.6.1 Four-tier system architecture

We chose this architecture mostly for portability: it decouples the client implementation from the business logic so it is easier to support a vast number of client types.

A light client also means it can be supported on a vast variety of even older hardware. Other benefit from this choice are:

- *Flexibility*: the communication through known interfaces makes the modification of single components easier without the need to check the entire system functionality.
- *Scalability*: the separation of layers makes it easier to expand just the critical components, saving money and time on less critical parts of the system.
- *Fault tolerance*: the presence of redundant servers guarantees functionality in the case of high load or some server fault.
- *Load distribution*: the load balancer could be dynamically implemented to preview high spikes of load and dynamically assign and free computation power from the critical components. Especially useful if it is deployed in a server farm.

2.6.2 RESTful architecture

The choice of the RESTful architecture to communicate to the application server brings many advantages:

- *Scalability, flexibility and portability*: the stateless nature of the server leads to an easier maintenance, meaning that a server can be migrated or added without any changes to data.
- *Independence*: the greater separation between the presentation and application layer makes the development of clients easier, so they can be built independently from the rest of the system.

2.6.3 Model View Controller

Model-View-Controller is a design pattern used to divide the program into three interconnected elements. It is useful to separate the internal representation of information from how the information is then presented to the user. The three components are:

- *Model*: the system internal dynamic data structure. It directly manages data and rules of the application. It keeps the internal state consistent.
- *Controller*: accepts input from the view and applies elaboration to the model.
- *View*: representation of the internal data for the user. Multiple views of the same information are possible.

2.7 Other design decisions

2.7.1 Thin Client

A thin client only has to display information on the screen. We chose this design for our system to be able to bring our application to the highest variety of devices possible. This eases the work of porting the application to new systems and guarantees that even low powered devices can run the application properly. The drawback of this choice is that we need to add the requirement for an internet connection.

2.7.2 Stateless components

The server components being stateless enables our system to be highly flexible and ease the work for scaling out, enabling the possibilities of an elastic load balancer to dynamically manage resources (and reduce costs). Adding new compute units is easy as they don't need to copy or import any data, but they just need to execute the components. Also reliability of the system is increased as if a component fails no data will be lost and it can just be easily substituted.

3 User Interface Design

3.1 User

- Account creation

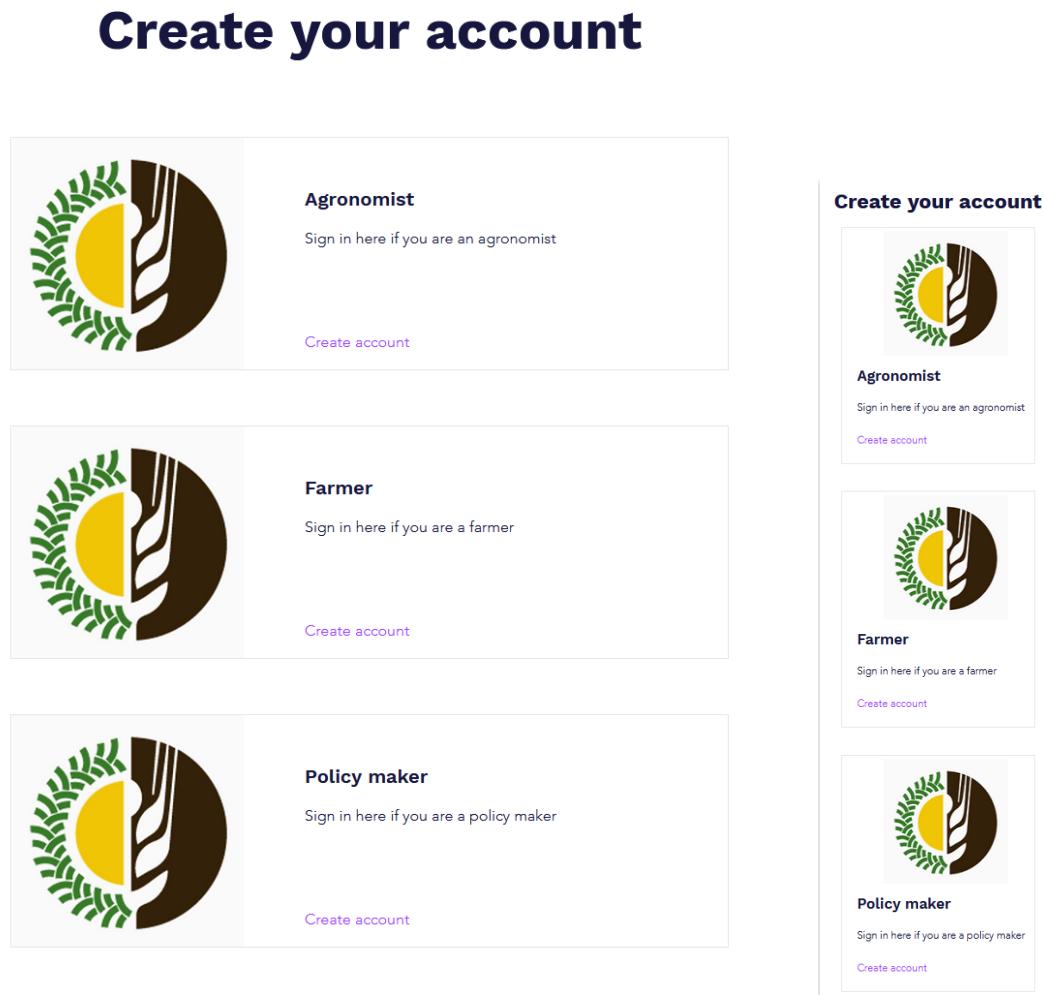


Figure 20: Create Account; web on the left, app on the right

- First page after download

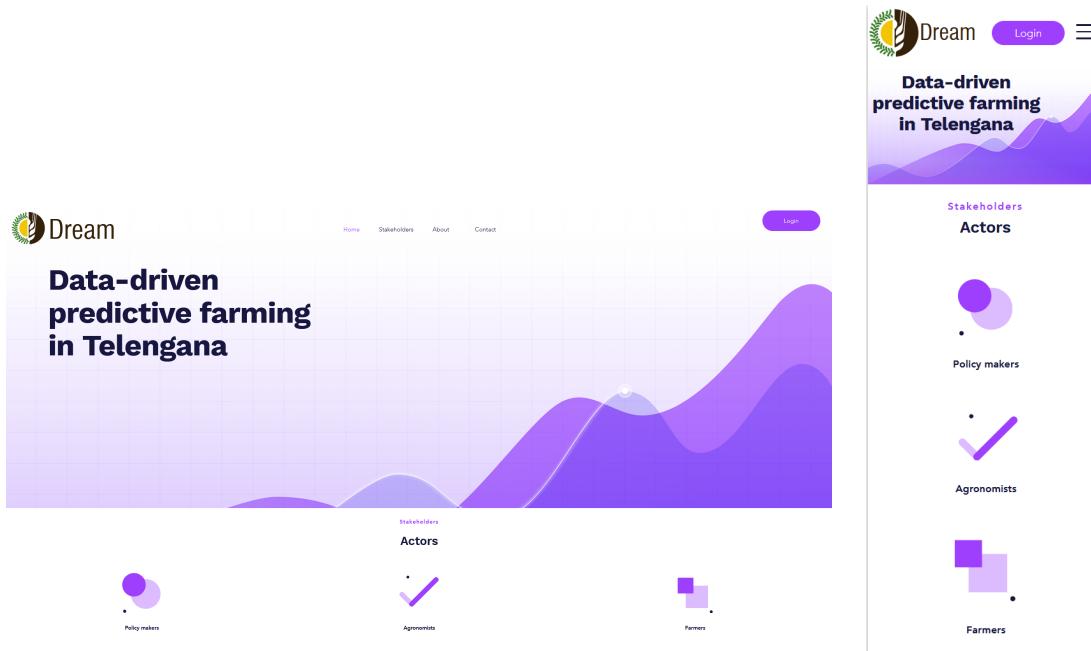


Figure 21: First page; web on the left, app on the right

- User Login

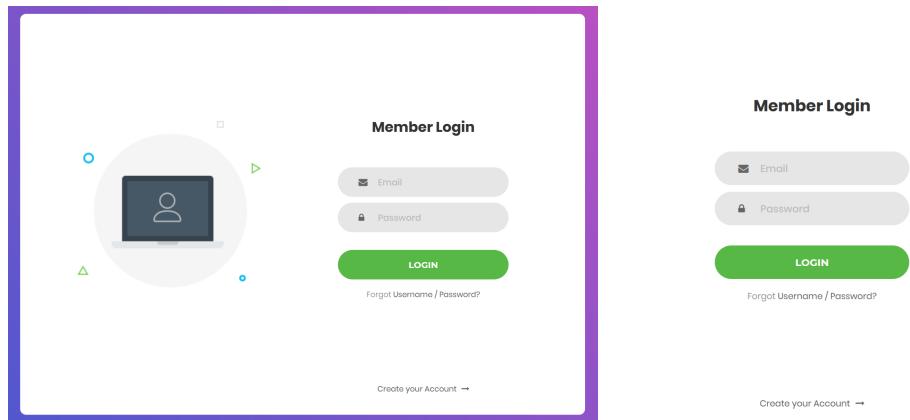


Figure 22: User Login; web on the left, app on the right

3.2 Farmer

- Farmer Sign In

Create your account

Farmer

Sign in here if you are a farmer



Name: <<Insert your name>>
Surname: <<Insert your surname>>
Email: <<Insert your email>>
Password: <<Insert your password>>
Farm's name: <<Insert farm's name>>
Farm's location: <<Insert farm's location>>

By signing in you accept the [Terms of Services](#)

Confirm

Create your account

Farmer

Sign in here if you are a farmer



Name: <<Insert your name>>
Surname: <<Insert your surname>>
Password: <<Insert your password>>
Email: <<Insert your email>>
Farm's name: <<Insert farm's name>>
Farm's location: <<Insert farm's location>>

By signing in you accept the [Terms of Services](#)

Confirm

Figure 23: Farmer Sign In; web on the left, app on the right

- Farmer Home Page

The image displays two versions of the Dream Farmer Home Page: a web version on the left and a mobile application version on the right.

Web Version (Left):

- Header:** Dream logo with a stylized leaf icon.
- Navigation:** Home, Weather forecasts, Requests, Forums, Production&Problems.
- Main Content:**
 - A large banner with the text "Solutions for a healthier future" and an image of a yellow combine harvester in a field.
 - A section titled "Weather forecasts for today" showing the current weather in Hyderabad, Telangana: 26° Haze.
 - A detailed "Today's Forecast for Hyderabad, Telangana" table:

Time	Temperature	Humidity
Morning	26°	---
Afternoon	26°	0%
Evening	21°	1%
Overnight	19°	5%
- Mobile Version (Right):**
- Header:** Dream logo with a stylized leaf icon and a menu icon.
- Slogan:** "Solutions for a healthier future" with an image of a combine harvester.
- Section:** "Weather forecasts for today" showing the current weather in Hyderabad, Telangana: 26° Haze.
- Section:** "Personalized suggestions" featuring a photo of young green plants and a link to "Fertilizer rates by placement approach".
- Text:** A paragraph about reduced phosphorus and potassium rates for corn production when applied in a band.
- Section:** "Personalized suggestions" featuring a photo of young green plants and a link to "Fertilizer rates by placement approach".
- Text:** A paragraph about reduced phosphorus and potassium rates for corn production when applied in a band.

Figure 24: Farmer Home Page; web on the left, app on the right

- Farmer Best Practices



The screenshot shows the Dream web application's interface. At the top, there is a navigation bar with links for Home, Weather forecasts, Requests, Forums, and Production&Problems. The main content area has a brown header with the text "Best practices". On the left, there is a circular logo featuring a stylized sun and leaves. Below the logo, the text "Insert best practices" is displayed, followed by a placeholder "=<Insert here your suggestions>=". A "Send" button is located at the bottom right of this section. The main body of the page contains a message from a user named Kamalakar, which reads:

Hi I'm Kamalakar, a policy maker who has noticed the great work you've done recently.
I've seen that despite all the meteorological adverse events you've encountered during the last year, your production has increased. Do you have any suggestions that farmers in your area could follow?

Below this message, there is another "Insert best practices" section with a "Send" button.



The screenshot shows the Dream mobile application's interface. It features a similar layout to the web version, with a brown header and a "Best practices" section. The user message from Kamalakar is identical to the one on the web version. Below the message, there is another "Insert best practices" section with a "Send" button.

Figure 25: Farmer inserts Best Practices; web on the left, app on the right

- Farmer Forum

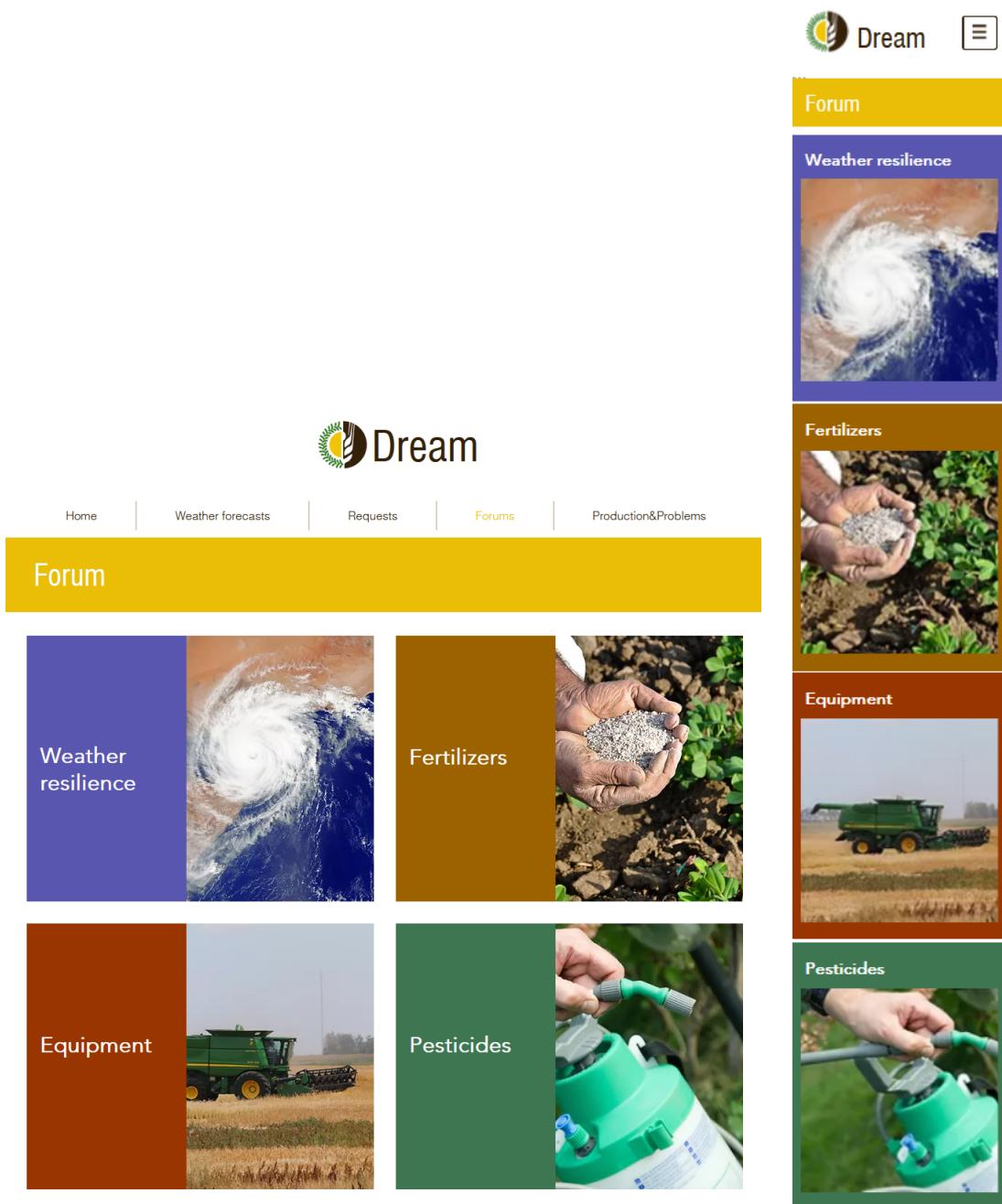


Figure 26: Farmer Forum Home Page; web on the left, app on the right

The image displays two views of a farmer forum page for the 'Fertilizers' topic. On the left is the web version, and on the right is the mobile app version. Both versions feature a header with the 'Dream' logo and navigation links for Home, Weather forecasts, Requests, Forums, and Production&Problems. A search bar is present in both.

Web Version (Left):

- Fertilizers Topic:** Shows a photo of hands holding fertilizer granules over a field. Below it is the title 'Fertilizers'.
- Discussion Metrics:** Discussions: 122, Messages: 338. Includes a 'New post' button.
- Subscribe Button:** A yellow 'Subscribe' button.
- Topic Cards:**
 - Fertilizer rates:** Answers: 23, Visits: 87, Last message: 21/12/2021.
 - Pre-Planting Fertilizers:** Answers: 14, Visits: 23, Last message: 18/12/2021.
 - When should you fertilize?** Answers: 8, Visits: 15, Last message: 15/12/2021.

Mobile App Version (Right):

- Fertilizers Topic:** Shows a photo of hands holding fertilizer granules over a field. Below it is the title 'Fertilizers'.
- Discussion Metrics:** Discussions: 122, Messages: 338. Includes a 'New post' button.
- Subscribe Button:** A yellow 'Subscribe' button.
- Topic Cards:**
 - Fertilizer rates:** Answers: 23, Visits: 87, Last message: 21/12/2021.
 - Pre-Planting Fertilizers:** Answers: 14, Visits: 23, Last message: 18/12/2021.
 - When should you fertilize?** Answers: 8, Visits: 15, Last message: 15/12/2021.

Figure 27: Farmer Forum Page Subscribe; web on the left, app on the right

Fertilizers

Discussions: 122 Messages: 338 [New post](#) NEW

[Unsubscribe](#)

Fertilizer rates

Answers: 23 Visits: 87 Last message: 21/12/2021

Pre-Planting Fertilizers

Answers: 14 Visits: 23 Last message: 18/12/2021

When should you fertilize?

Answers: 8 Visits: 15 Last message: 15/12/2021

Fertilizers

Discussions: 122 Messages: 338 [New post](#) NEW

[Unsubscribe](#)

Fertilizer rates

Answers: 23 Visits: 87 Last message: 21/12/2021

Pre-Planting Fertilizers

Answers: 14 Visits: 23 Last message: 18/12/2021

When should you fertilize?

Answers: 8 Visits: 15 Last message: 15/12/2021

Figure 28: Farmer Forum Page Unsubscribe; web on the left, app on the right

The image shows two side-by-side views of a mobile application interface for a farmer forum. Both views feature a header with the 'Dream' logo and navigation links for 'Home', 'Weather forecasts', 'Requests', 'Forums', 'Production&Problems', and a menu icon.

Left View (Web Version):

- Title:** <<Insert here the title of the post>>
- Post:** <<Insert here the text of the post>>
- Image:** A photograph of a person's hands holding a handful of dark, granular fertilizer over a field of young green plants.
- Text Overlay:** The word "Fertilizers" is overlaid in large white capital letters across the bottom of the image.
- Send Button:** A yellow rounded rectangle button labeled "Send" with a right-pointing arrow.

Right View (App Version):

- Title:** <<Insert here the title of the post>>
- Post:** <<Insert here the text of the post>>
- Image:** A photograph of a person's hands holding a handful of dark, granular fertilizer over a field of young green plants.
- Text Overlay:** The word "Fertilizers" is overlaid in large white capital letters across the bottom of the image.
- Send Button:** A yellow rounded rectangle button labeled "Send" with a right-pointing arrow.

Figure 29: Farmer Forum New Post; web on the left, app on the right

The screenshot shows a web-based farmer forum interface. At the top, there's a navigation bar with links for Home, Weather forecasts, Requests, Forums, and Production&Problems. Below the navigation is a large image of a person's hands holding fertilizer granules over a field of young plants. Overlaid on the image is the word "Fertilizers". To the right of the image, the title "Fertilizer rates" is displayed. A question from user Ajitabh is shown: "I've recently noticed that my fertilizer rates are too high. How can I reduce the amount without reducing its effect too?". Below the question are the message source ("Message From: Ajitabh (Ajitabh.86@gmail.com)"), the time ("Time: 21/12/2021: 21.05"), and a yellow "Answer" button.

Answers: 23
Visits: 87
Last message: 21/12/2021

Answers

Response from: Devilal (Devilal@gmail.com)
Time: 21/12/2021 - 21:05
As already told before you can use reduced phosphorus and potassium rates if you apply it in a band, thanks to the method's improved efficiency for corn.

Response from: Harish (Harish@gmail.com)
Time: 21/12/2021 - 19:31
Try applying it in bands.

Response from: Gurudas (Gurudas@gmail.com)
Time: 21/12/2021 - 08:13



Figure 30: Farmer Forum Specific Post; web on the left, app on the right



Home | Weather forecasts | Requests | Forums | Production&Problems



Fertilizer rates

I've recently noticed that my fertilizer rates are too high. How can I reduce the amount without reducing its effect too?

Message From: Ajitabh
(Ajitabh.86@gmail.com)
Time: 21/12/2021 - 21.05



Fertilizer rates

I've recently noticed that my fertilizer rates are too high. How can I reduce the amount without reducing its effect too?

Message From: Ajitabh
(Ajitabh.86@gmail.com)
Time: 21/12/2021 - 21.05

Answer:
<<Insert here the text of the response>>

Send →

Send →

Figure 31: Farmer Forum Response in a Post; web on the left, app on the right

- Farmer Help and Suggestions

The screenshot shows the Dream app's Requests section. At the top, there is a navigation bar with links for Home, Weather forecasts, Requests, Forums, and Production&Problems. Below the navigation bar, a large dark header bar displays the word "Requests". On the left side of the main content area, there is a message: "Here's your request history". The main content area contains three request entries:

- Fertilizer rates - 21/12/2021: 21:05**
I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?
- Backhoe, buy or rent? - 18/12/2021: 18:27**
Is it better to buy a backhoe (even non a new one) or to rent it? Any advice?
- Atrazine seller - 16/12/2021: 7:34**
I regularly use Atrazine to eliminate broadleaved weeds, but the price has recently increased quite a lot. Do you know any seller with a fair price?

On the right side of the main content area, there is a yellow button labeled "Create new request →". Above the main content area, there is a smaller section titled "Requests" with a "Create new request" button.

Figure 32: Farmer Help and Suggestions Request History; web on the left, app on the right

The image displays two versions of a 'Create new request' form: a desktop web version on the left and a mobile app version on the right.

Web Version (Left):

- Header:** Dream logo, navigation menu (Home, Weather forecasts, Requests, Forums, Production&Problems), and a 'Create new request' button.
- Form Fields:**
 - Description:** Text area with placeholder text: '<<Insert here the description of your request>>'.
 - Image Preview:** A small image of a green plant.
 - Addressee:** Section with four options:
 - All farmers:** Icon of a windmill and sun, description: 'Send the request to all the farmers of your area'.
 - All agronomists:** Icon of two people, description: 'Send the request to all the sronomist of your area'.
 - Specific farmer:** Icon of a person working in a field, description: '<<Click here to insert the identifier of a farmer>>'.
 - Specific agronomist:** Icon of a person, description: '<<Click here to insert the identifier of an agronomist>>'.
 - Buttons:** 'Send' button at the bottom.

Figure 33: Farmer Help and Suggestions Create New Request; web on the left, app on the right

The screenshot shows the Dream app's user interface. At the top is a navigation bar with five items: Home, Weather forecasts, Requests, Forums, and Production&Problems. To the right of the navigation bar is the app's logo, which consists of a stylized green and yellow leaf-like icon followed by the word "Dream". Below the navigation bar, there is a section titled "Fertilizer rates" containing a message from a user named Ajitabh. The message reads: "I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?". Below the message are the sender's details: "Sender : Ajitabh (Ajitabh.86@gmail.com)" and "Time: 21/12/2021: 21.05". A yellow button labeled "Answer →" is positioned below the message. On the far right of the screen, there is a vertical menu icon consisting of three horizontal lines.

Fertilizer rates

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

[Answer →](#)

Fertilizer rates

I've recently noticed that my fertilizer rates are too high.

Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

[Answer →](#)

Figure 34: Farmer Help and Suggestions Specific Request; web on the left, app on the right

The screenshot shows the top navigation bar of the Dream website. It includes a logo with a stylized sun icon and the word "Dream". Below the logo are five menu items: "Home", "Weather forecasts", "Requests", "Forums", and "Production&Problems".

Fertilizer rates

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

Response

<<Insert here your response>>

Send →

The screenshot shows the mobile application interface for the Dream platform. It features a header with the "Dream" logo and a menu icon. Below the header, the "Fertilizer rates" request is displayed, along with the sender's information and a response input field.

Fertilizer rates

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

Response

<<Insert here your response>>

Send →

Figure 35: Farmer Help and Suggestions Response to a Specific Request; web on the left, app on the right

- Farmer Personalized Suggestions

The screenshot displays the Dream web application interface. At the top, there's a navigation bar with links for Home, Weather forecasts, Requests, Forums, and Production&Problems. Below this, a large dark banner reads "Personalized suggestions". Underneath, a sub-section titled "Here you can find some suggestions that may help you" lists several items:

- Fertilizers**: Fertilizer rates by placement approach. Description: You can use reduced phosphorus and potassium rates if you apply it in a band, thanks to the method's improved efficiency for corn. Tables 3 and 4 show the difference in recommended rates as affected by band and broadcast placement, respectively, for corn production. Using lower rates...
- Equipment**: Backhoe. Description: If digging is your thing, a backhoe is your tool; if you don't plan to dig holes on a regular basis, you'd be better served to borrow or rent a backhoe rather than purchase your own...
- Logistics**: Management. Description: It is important to organize accurately your logistic with seed to table supply-chain management, and financial management,...
- Pesticides**: Atrazine. Description: Atrazine can be applied to eliminate broadleaved weeds preventing photosynthesis in them, and making them starve. It is inexpensive, does not damage crops heavily and can be applied during a long window of time...

Figure 36: Farmer Personalized Suggestions; web on the left, app on the right



Home | Weather forecasts | Requests | Forums | Production&Problems



Fertilizer rates by placement approach

You can use reduced phosphorus and potassium rates if you apply it in a band, thanks to the method's improved efficiency for corn. Tables 3 and 4 show the difference in recommended rates as affected by band and broadcast placement, respectively, for corn production. Using lower rates of phosphorus and/or potassium in a band reduces the amount of money spent on phosphate and/or potash fertilizers.

Table 3: Phosphate guidelines for corn production								
Yield goal	Broadcast, very low soil test P	Band, very low soil test P	Broadcast, low soil test P	Band, medium soil test P	Broadcast, high soil test P	Band, high soil test P	Broadcast, very high soil test P	
70-90 bushels per acre	50 lbs. of P2O5	25 lbs. of P2O5	35 lbs. of P2O5	20 lbs. of P2O5	15 lbs. of P2O5	10 lbs. of P2O5	10-15 lbs. of P2O5	
91-110 bushels per acre	60 lbs. of P2O5	30 lbs. of P2O5	40 lbs. of P2O5	20 lbs. of P2O5	25 lbs. of P2O5	20 lbs. of P2O5	10 lbs. of P2O5	
111-130 bushels per acre	75 lbs. of P2O5	40 lbs. of P2O5	50 lbs. of P2O5	25 lbs. of P2O5	30 lbs. of P2O5	20 lbs. of P2O5	10 lbs. of P2O5	
131-150 bushels per acre	85 lbs. of P2O5	45 lbs. of P2O5	60 lbs. of P2O5	30 lbs. of P2O5	35 lbs. of P2O5	25 lbs. of P2O5	10 lbs. of P2O5	

Use one of the following equations if you want a P₂O₅ recommendation for a specific soil test value and a specific yield goal:

- Recommended P₂O₅ = [0.700 - .035 (Bray P ppm)] (yield goal)
- Recommended P₂O₅ = [0.700 - .044 (Olsen P ppm)] (yield goal)

No phosphate fertilizer is recommended if the soil test for P is higher than 25 parts per million (ppm) with the Bray test or 20 ppm with the Olsen test.



Fertilizer rates by placement approach
You can use reduced phosphorus and potassium rates if you apply it in a band, thanks to the method's improved efficiency for corn. Tables 3 and 4 show the difference in recommended rates as affected by band and broadcast placement, respectively, for corn production. Using lower rates of phosphorus and/or potassium in a band reduces the amount of money spent on phosphate and/or potash fertilizers.

Table 3: Phosphate guidelines for corn production								
Yield goal	Broadcast, very low soil test P	Band, very low soil test P	Broadcast, low soil test P	Band, medium soil test P	Broadcast, high soil test P	Band, high soil test P	Broadcast, very high soil test P	
70-90 bushels per acre	50 lbs. of P2O5	25 lbs. of P2O5	35 lbs. of P2O5	20 lbs. of P2O5	15 lbs. of P2O5	10 lbs. of P2O5	10-15 lbs. of P2O5	
91-110 bushels per acre	60 lbs. of P2O5	30 lbs. of P2O5	40 lbs. of P2O5	20 lbs. of P2O5	25 lbs. of P2O5	20 lbs. of P2O5	10 lbs. of P2O5	
111-130 bushels per acre	75 lbs. of P2O5	40 lbs. of P2O5	50 lbs. of P2O5	25 lbs. of P2O5	30 lbs. of P2O5	20 lbs. of P2O5	10 lbs. of P2O5	
131-150 bushels per acre	85 lbs. of P2O5	45 lbs. of P2O5	60 lbs. of P2O5	30 lbs. of P2O5	35 lbs. of P2O5	25 lbs. of P2O5	10 lbs. of P2O5	

Use one of the following equations if you want a P₂O₅ recommendation for a specific soil test value and a specific yield goal:

- Recommended P₂O₅ = [0.700 - .035 (Bray P ppm)] (yield goal)
- Recommended P₂O₅ = [0.700 - .044 (Olsen P ppm)] (yield goal)

No phosphate fertilizer is recommended if the soil test for P is higher than 25 parts per million (ppm) with the Bray test or 20 ppm with the Olsen test.

Figure 37: Farmer Specific Personalized Suggestions; web on the left, app on the right

- Farmer Production and Problems

The image shows a comparison between a web-based platform and a mobile application for farmers. On the left, the web interface features a header with a logo and navigation links for Home, Weather forecasts, Requests, Forums, and Production&Problems. Below the header is a section titled "Production & Problems" with a sub-instruction: "Here you can either insert your production data or report a problem you are facing in your production to an agronomist". This section includes two large buttons: one with a wheat field background labeled "Insert production" and another with a teal background labeled "Report problem". On the right, the mobile application interface shows a similar header with the "Dream" logo and a menu icon. It also has a "Production & Problems" section with the same instructions and "Insert production" and "Report problem" buttons. The mobile app's design is more compact and focused on the main functions.

Figure 38: Farmer Production and Problems; web on the left, app on the right



Home

Weather forecasts

Requests

Forums

Production&Problems

Production history

Insert data →

December 2021

Amount of corn produced = 1750 Kg
Amount of energy used per kg = 2MJ
Amount of fertilizer used per unit = 50 kg
Amount of water used per unit = 2 625 000 l



December 2020

Amount of corn produced = 1250 Kg
Amount of energy used per kg = 1.8 MJ
Amount of fertilizer used per unit = 60 kg
Amount of water used per unit = 2 700 000 l



Production history

Insert data →

December 2021

Amount of corn produced = 1750 Kg
Amount of energy used per kg = 2MJ
Amount of fertilizer used per unit = 50 kg
Amount of water used per unit = 2 625 000 l



December 2020

Amount of corn produced = 1250 Kg
Amount of energy used per kg = 1.8 MJ
Amount of fertilizer used per unit = 60 kg
Amount of water used per unit = 2 700 000 l



Figure 39: Farmer Production History; web on the left, app on the right



Home | Weather forecasts | Requests | Forums | Production&Problems

Insert production data



Amount of corn produced

<<Insert the amount of corn produced>>



Amount of energy used per unit

<<Insert the amount of energy used per unit>>



Amount of fertilizer used per unit

<<Insert here the amount of fertilizer used per unit>>



Amount of water used per unit

<<Insert here the amount of water used per unit>>

Save →



≡



Insert production data



Amount of corn produced

<<Insert the amount of corn produced>>



Amount of energy used per unit

<<Insert the amount of energy used per unit>>



Amount of fertilizer used per unit

<<Insert here the amount of fertilizer used per unit>>

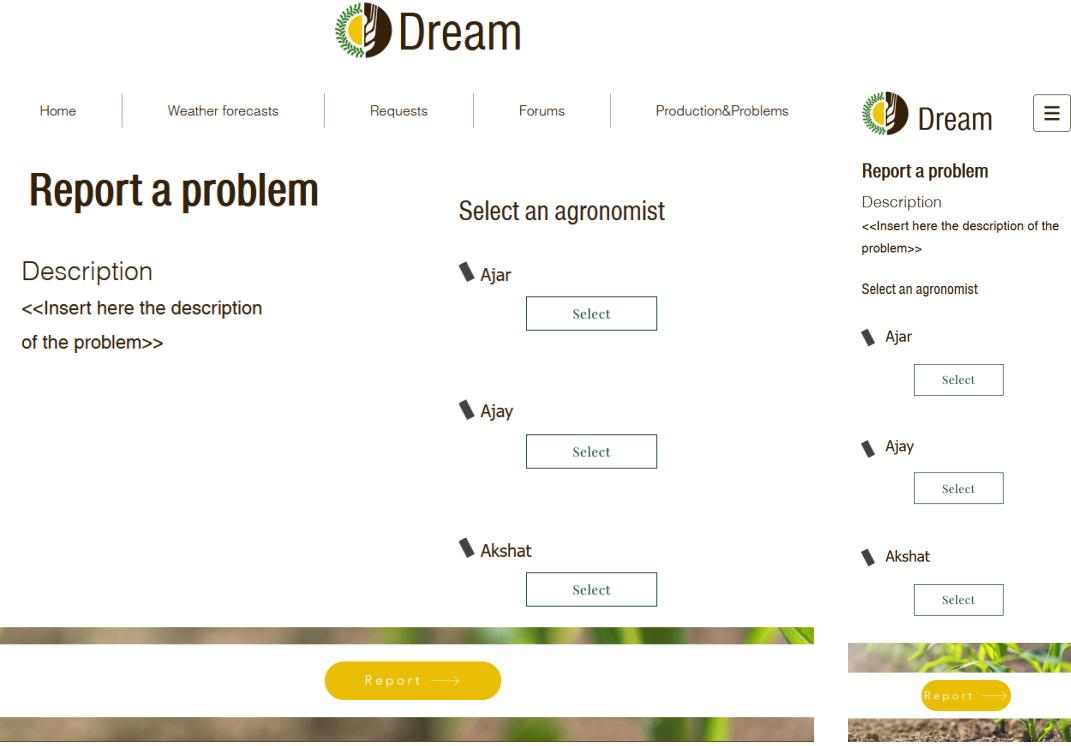


Amount of water used per unit

<<Insert here the amount of water used per unit>>

Save →

Figure 40: Farmer Insert Production Data; web on the left, app on the right



The figure shows a comparison between a web-based interface and a mobile application interface for reporting a problem.

Web Interface (Left):

- Header:** Dream logo.
- Navigation:** Home, Weather forecasts, Requests, Forums, Production&Problems.
- Title:** Report a problem.
- Description:** A text input field with placeholder text: <<Insert here the description of the problem>>.
- Select an agronomist:** A section with three options: Ajay, Ajay, and Akshat, each with a "Select" button.
- Report Button:** A yellow button labeled "Report" with an arrow pointing right.

Mobile Application Interface (Right):

- Header:** Dream logo with a menu icon.
- Title:** Report a problem.
- Description:** A text input field with placeholder text: <<Insert here the description of the problem>>.
- Select an agronomist:** A section with three options: Ajay, Ajay, and Akshat, each with a "Select" button.
- Report Button:** A yellow button labeled "Report" with an arrow pointing right.

Figure 41: Farmer Report a Problem; web on the left, app on the right

- Farmer Weather

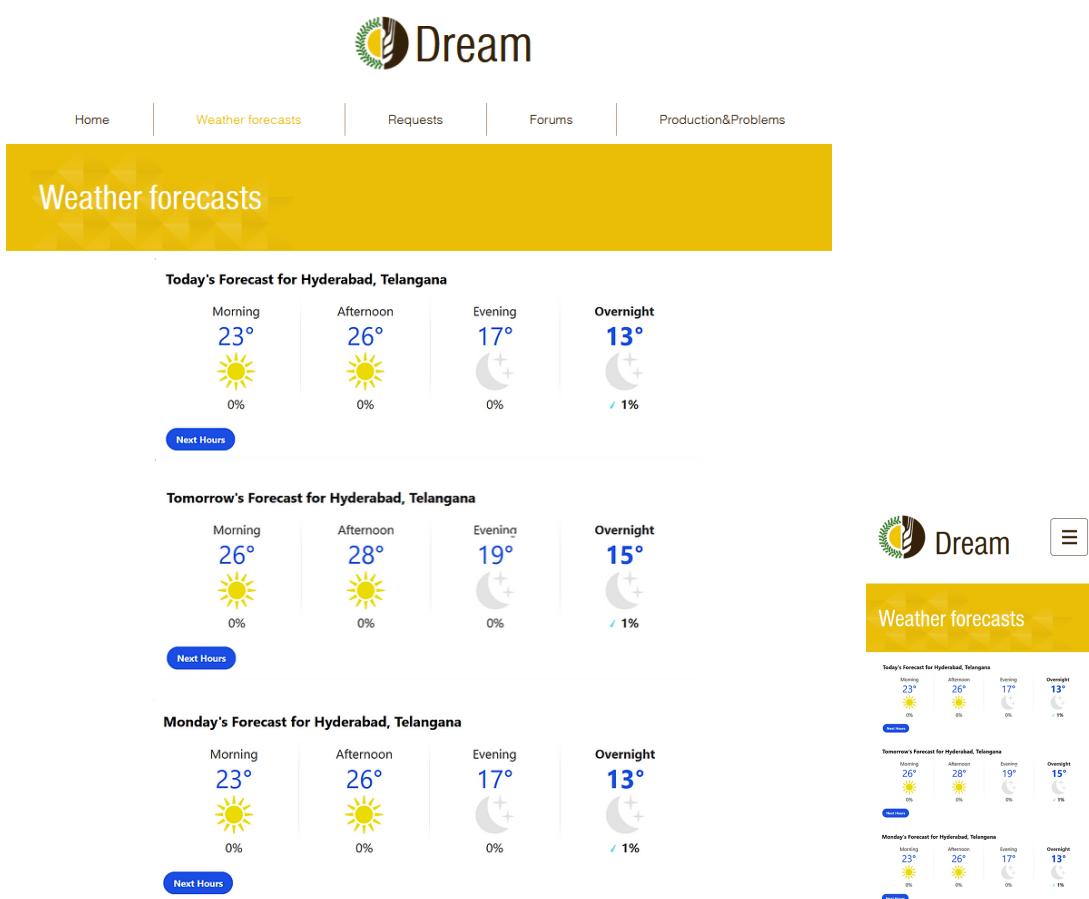


Figure 42: Farmer Days Forecasts; web on the left, app on the right



Home | Weather forecasts | Requests | Forums | Production&Problems

Tomorrow's forecast



Hourly Weather - Hyderabad, Telangana

As of 03:03 IST

Tuesday, 21 December

03:00	14°		Clear	1% NNW 0 km/h	
04:00	13°		Clear	1% N 0 km/h	
05:00	13°		Clear	1% NW 0 km/h	
06:00	12°		Clear	1% NW 0 km/h	
07:00	13°		Sunny	1% NW 0 km/h	
08:00	17°		Sunny	1% NNE 0 km/h	
09:00	20°		Sunny	0% NNE 1 km/h	
10:00	23°		Sunny	0% NNE 1 km/h	
11:00	24°		Sunny	0% NNE 1 km/h	
12:00	25°		Sunny	0% NNW 2 km/h	
13:00	26°		Sunny	0% NNW 3 km/h	
14:00	26°		Sunny	0% NNW 3 km/h	
15:00	26°		Sunny	0% NNW 4 km/h	
16:00	26°		Sunny	0% NNW 3 km/h	
17:00	24°		Cloudy	0% N 3 km/h	



Tomorrow's forecast



Figure 43: Farmer Hours Forecasts; web on the left, app on the right

3.3 Agronomist

- Agronomist Sign In

Create your account

Agronomist

Sign in here if you are an agronomist



Name: <<Insert your name>>

Surname: <<Insert your surname>>

Email: <<Insert your email>>

Password: <<Insert your password>>

Responsible area: <<Insert responsible area>>

By signing in you accept the [Terms of Services](#)

Confirm

Create your account

Agronomist

Sign in here if you are an agronomist



Name: <<Insert your name>>

Surname: <<Insert your surname>>

Password: <<Insert your password>>

Email: <<Insert your email>>

Responsible area: <<Insert responsible area>>

By signing in you accept the [Terms of Services](#)

Confirm

Figure 44: Agronomist Sign In; web on the left, app on the right

- Agronomist Home Page

The figure displays two versions of the Agronomist Home Page: a web interface on the left and a mobile application interface on the right, both branded under the 'Dream' logo.

Web Interface (Left):

- Header:** Dream
- Navigation:** Home, Farmer's performance, Weather forecasts, Requests, Daily plan
- Hero Image:** A yellow combine harvester in a field.
- Section: Weather forecasts for today**
 - Current Weather:** Hyderabad, Telangana Weather As of 16:36 IST, 26° Haze
 - Today's Forecast:**

Morning	Afternoon	Evening	Overnight
26°	26°	21°	19°
Waves icon	Waves icon	Cloud icon	Cloud icon
0%	0%	1%	5%
- Section: Today's plan**
 - To Do:** A whiteboard with "To Do:" written on it.
 - Tasks:**
 - 8.00: Standard control Basdev's farm
 - 9.00: Check Jaidev's wind resilience
 - 11.00: Standard control Mahavir's farm
 - 14.00: Check Prabir's improvements
 - ...
 - Read More** button

The mobile application interface shows a compact version of the web content, including the header, navigation, weather forecast, and a simplified 'Today's plan' section.

- Header:** Dream
- Section: Solutions for a healthier future**
- Section: Weather forecasts for today**
 - Current Weather:** Hyderabad, Telangana Weather As of 16:36 IST, 26° Haze
 - Today's Forecast:**

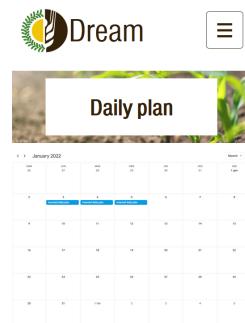
Morning	Afternoon	Evening	Overnight
26°	26°	21°	19°
Waves icon	Waves icon	Cloud icon	Cloud icon
0%	0%	1%	5%
- Section: Today's plan**
 - To Do:** A whiteboard with "To Do:" written on it.
 - Tasks:**
 - 8.00: Standard control Basdev's farm
 - 9.00: Check Jaidev's wind resilience
 - 11.00: Standard control Mahavir's farm
 - 14.00: Check Prabir's improvements
 - ...
 - Read More** button

Figure 45: Agronomist Home Page; web on the left, app on the right

- **Agronomist Daily Plan**



The web version of the Agronomist Daily Plan features a header with the "Dream" logo and navigation links for "Home", "Farmer's performance", "Weather forecasts", "Requests", and "Daily plan". Below the header is a large image of young corn plants. A white rectangular overlay in the center contains the text "Daily plan". Below the image is a calendar for January 2022. The days of the week are labeled in Italian: DOM, LUN, MAR, MER, GIO, VEN, SAB. The dates range from 26 to 31 January, with 1 February listed as the next day. The days 3, 4, and 5 are highlighted with blue backgrounds and the text "Inserted daily plan" below them.



The app version of the Agronomist Daily Plan has a similar layout but is presented as a mobile application interface. It includes a header with the "Dream" logo and a menu icon. Below the header is a smaller image of young corn plants with a "Daily plan" overlay. The main area shows the same January 2022 calendar grid, with the days 3, 4, and 5 highlighted in blue with the "Inserted daily plan" message.

Figure 46: Agronomist Daily Plan Home Page; web on the left, app on the right

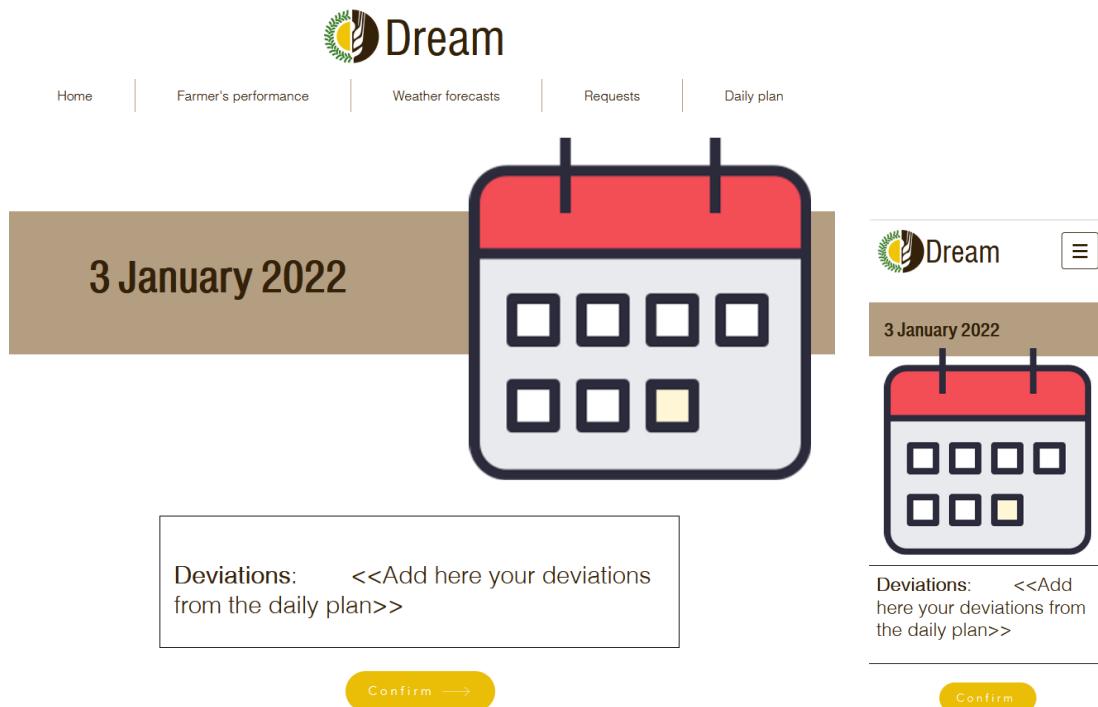


Figure 47: Agronomist Confirm Daily Plan or Specify Deviations; web on the left, app on the right

The figure displays two side-by-side interfaces for agronomists, both titled "Dream".

Left (Web Interface):

- Header:** Home | Farmer's performance | Weather forecasts | Requests | Daily plan.
- Date:** 3 January 2022
- Central Element:** A large calendar icon showing a grid of squares. The top row has four squares, the second row has three, and the third row has two, with the last square being yellow.
- Section Headers:** Plan
- Events:**
 - 8.00-9.00 Standard control Basdev's farm
 - 9.00-11.00 Check Jaidev's wind resilience
 - 11.00-14.00 Standard control Mahavir's farm
 - 15.00-17.00 Check Prabir's improvements
- Buttons:** Confirm/Specify deviations →

Right (Mobile App Interface):

- Header:** Dream | ☰
- Date:** 3 January 2022
- Central Element:** A smaller calendar icon showing a grid of squares, with the last square being yellow.
- Section Headers:** Plan
- Events:**
 - 8.00-9.00 Standard control Basdev's farm
 - 9.00-11.00 Check Jaidev's wind resilience
 - 11.00-14.00 Standard control Mahavir's farm
 - 15.00-17.00 Check Prabir's improvements
- Buttons:** Confirm/Specify deviations

Figure 48: Agronomist Specific Day Confirm Daily Plan or Specify Deviations; web on the left, app on the right

The figure shows two views of a daily plan for January 3, 2022.

Left (Web View):

- Header:** Home, Farmer's performance, Weather forecasts, Requests, Daily plan.
- Date:** 3 January 2022.
- Calendar:** A large calendar icon showing the date 3 January 2022.
- Plan:** A list of scheduled tasks:
 - 8.00-9.00 Standard control Basdev's farm
 - 9.00-11.00 Check Jaidev's wind resilience
 - 11.00-14.00 Standard control Mahavir's farm
 - 15.00-17.00 Check Prabir's improvements
- Buttons:** Confirmed daily plan.

Right (App View):

- Header:** Dream, menu icon.
- Date:** 3 January 2022.
- Calendar:** A smaller calendar icon showing the date 3 January 2022.
- Plan:** A list of scheduled tasks:
 - 8.00-9.00 Standard control Basdev's farm
 - 9.00-11.00 Check Jaidev's wind resilience
 - 11.00-14.00 Standard control Mahavir's farm
 - 15.00-17.00 Check Prabir's improvements
- Buttons:** Confirmed daily plan.

Figure 49: Agronomist Confirmed Execution of Daily Plan; web on the left, app on the right

The figure displays two side-by-side interfaces for managing a daily plan. Both interfaces feature a large calendar icon with a red header and a grey body containing four rows of four squares each. The top-left square of the second row is highlighted in yellow. Below the calendar is the word "Plan".

Web Interface (Left):

- Header: Dream, Home, Farmer's performance, Weather forecasts, Requests, Daily plan.
- Date: 3 January 2022.
- Form fields:
 - Start time: <<Insert start time>>
 - End time: <<Insert end time>>
 - Description: <<Insert description>>
- Buttons: Add activity (green), Save (yellow).

Mobile App Interface (Right):

- Header: Dream, menu icon.
- Date: 3 January 2022.
- Form fields:
 - Start time: <<Insert start time>>
 - End time: <<Insert end time>>
 - Description: <<Insert description>>
- Buttons: Add activity (green), Save (yellow).

Figure 50: Agronomist Insert Daily Plan; web on the left, app on the right

The screenshot displays the Dream agronomist daily plan interface. At the top, there is a navigation bar with links: Home, Farmer's performance, Weather forecasts, Requests, and Daily plan. The main area shows a calendar for January 3, 2022, with four tasks listed below it. The tasks are represented by boxes with start and end times and descriptions.

Start time	End time	Description
<<8.00>>	<<9.00>>	<<Standard control Basdev's farm>>
<<9.00>>	<<11.00>>	<<Check Jaidev's wind resilience>>
<<11.00>>	<<14.00>>	<<Standard control Mahavir's farm>>
<<15.00>>	<<17.00>>	<<Check Prabir's improvements>>

At the bottom right of the main area is a yellow "Save" button with a right-pointing arrow. To the right of the main interface is a smaller window showing the same calendar and task details, representing the mobile application version.

Figure 51: Agronomist Update Daily Plan; web on the left, app on the right

The figure displays two views of the Dream agronomist interface: a web-based view on the left and a mobile app view on the right.

Top Navigation: Both views include a top navigation bar with the 'Dream' logo and five menu items: Home, Farmer's performance, Weather forecasts, Requests, and Daily plan.

Calendar View: A large calendar icon for January 3, 2022, is centered. The calendar shows a grid of squares, with one square highlighted in yellow, indicating a task for that time slot.

Plan Section: Below the calendar, the word "Plan" is displayed. It lists four tasks with their respective times and descriptions:

- 8.00-9.00 Standard control Basdev's farm
- 9.00-11.00 Check Jaidev's wind resilience
- 11.00-14.00 Standard control Mahavir's farm
- 15.00-17.00 Check Prabir's improvements

Update Buttons: At the bottom of the web view, there is a large yellow "Update" button with a right-pointing arrow. In the app view, there are four smaller "Update" buttons, each located next to its corresponding task entry.

Figure 52: Agronomist Update Button; web on the left, app on the right

- Agronomist Visualize Farmers' performance

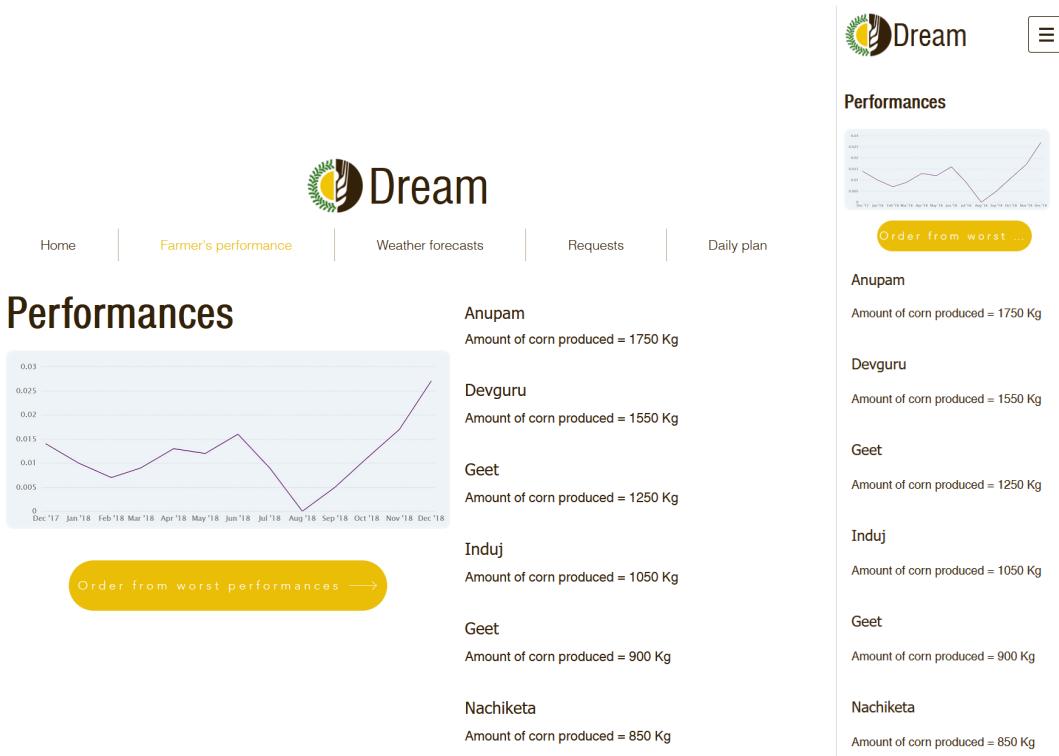


Figure 53: Agronomist Visualize Farmers' Performance; web on the left, app on the right

The screenshot displays the Dream agronomist web application interface. At the top, there is a navigation bar with links: Home, Farmer's performance, Weather forecasts, Requests, and Daily plan. Below the navigation bar is a large image of young corn plants growing in soil. Overlaid on this image is a white rectangular box containing the text "December 2021". To the right of the main content area, there is a sidebar with the Dream logo and a three-line menu icon. The sidebar contains the following information:

Farmer: Anupam

Amount of corn produced = 1750 Kg
Amount of energy used per kg = 2MJ
Amount of fertilizer used per unit = 50 kg
Amount of water used per unit = 2 625 000 l

Amount of corn produced =
1750 Kg
Amount of energy used per
kg = 2MJ
Amount of fertilizer used per
unit = 50 kg
Amount of water used per
unit = 2 625 000 l

Figure 54: Agronomist Visualize Specific Farmers' Performance; web on the left, app on the right

- **Agronomist Visualize Farmers' performance**

The screenshot displays two versions of the Dream agronomist requests interface side-by-side.

Left (Web Version):

- Header:** Dream logo with a stylized leaf icon.
- Navigation:** Home, Farmer's performance, Weather forecasts, Requests (highlighted in blue), Daily plan.
- Section:** Requests (dark blue header)
- Text:** "Here's your request history"
- Item 1:** **Fertilizer rates - 21/12/2021: 21:05**
I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?
- Item 2:** **Backhoe, buy or rent? - 18/12/2021: 18:27**
Is it better to buy a backhoe (even non a new one) or to rent it? Any advice?
- Item 3:** **Atrazine seller - 16/12/2021: 7:34**
I regularly use Atrazine to eliminate broadleaved weeds, but the price has recently increased quite a lot. Do you know any seller with a fair price?

Right (Mobile App Version):

- Header:** Dream logo with a stylized leaf icon and a menu icon.
- Section:** Requests (dark blue header)
- Text:** "Here's your request history"
- Item 1:** **Fertilizer rates - 21/12/2021: 21:05**
I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?
- Item 2:** **Backhoe, buy or rent? - 18/12/2021: 18:27**
Is it better to buy a backhoe (even non a new one) or to rent it? Any advice?
- Item 3:** **Atrazine seller - 16/12/2021: 7:34**
I regularly use Atrazine to eliminate broadleaved weeds, but the price has recently increased quite a lot. Do you know any seller with a fair price?

Figure 55: Agronomist Requests Home Page; web on the left, app on the right

The screenshot shows the Dream web application. At the top is a navigation bar with five items: "Home", "Farmer's performance", "Weather forecasts", "Requests", and "Daily plan". To the right of the "Requests" item is a menu icon consisting of three horizontal lines. Below the navigation bar, the main content area has a dark brown header bar. The main body contains a message from an agronomist:

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

A yellow button labeled "Answer →" is centered below the message.

The screenshot shows the Dream mobile application. At the top is a header with the "Dream" logo and a menu icon. Below the header, the main content area displays the same message from the agronomist:

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

A yellow button labeled "Answer →" is centered below the message.

Figure 56: Agronomist Specific Request; web on the left, app on the right

The screenshot shows the top navigation bar of the Dream web application. It includes a logo with a stylized leaf or plant icon followed by the word "Dream". Below the logo are five menu items: "Home", "Farmer's performance", "Weather forecasts", "Requests", and "Daily plan".

Fertilizer rates

I've recently noticed that my fertilizer rates are too high. Any suggestion on how to reduce the amount without reducing its effect too?

Sender : Ajitabh (Ajitabh.86@gmail.com)
Time: 21/12/2021: 21.05

Response

<<Insert here your response>>

Send →

The screenshot shows the mobile application interface for the Dream app. At the top is the "Dream" logo. Below it is a section titled "Fertilizer rates" containing the same message as the web version. Underneath is a "Response" section with the placeholder text "<<Insert here your response>>". A yellow "Send →" button is located at the bottom of this section.

Figure 57: Agronomist Respond to a Request; web on the left, app on the right

3.4 Policy Maker

- Policy Maker Sign In

Create your account

Policy maker

Sign in here if you are a policy maker



Name: <<Insert your name>>

Surname: <<Insert your surname>>

Email: <<Insert your email>>

Password: <<Insert your password>>

By signing in you accept the [Terms of Services](#)

Confirm

Create your account

Policy maker

Sign in here if you are a policy maker



Name: <<Insert your name>>

Surname: <<Insert your surname>>

Password: <<Insert your password>>

Email: <<Insert your email>>

By signing in you accept the [Terms of Services](#)

Confirm

Figure 58: Policy Maker Sign In; web on the left, app on the right

- Policy Maker Home Page

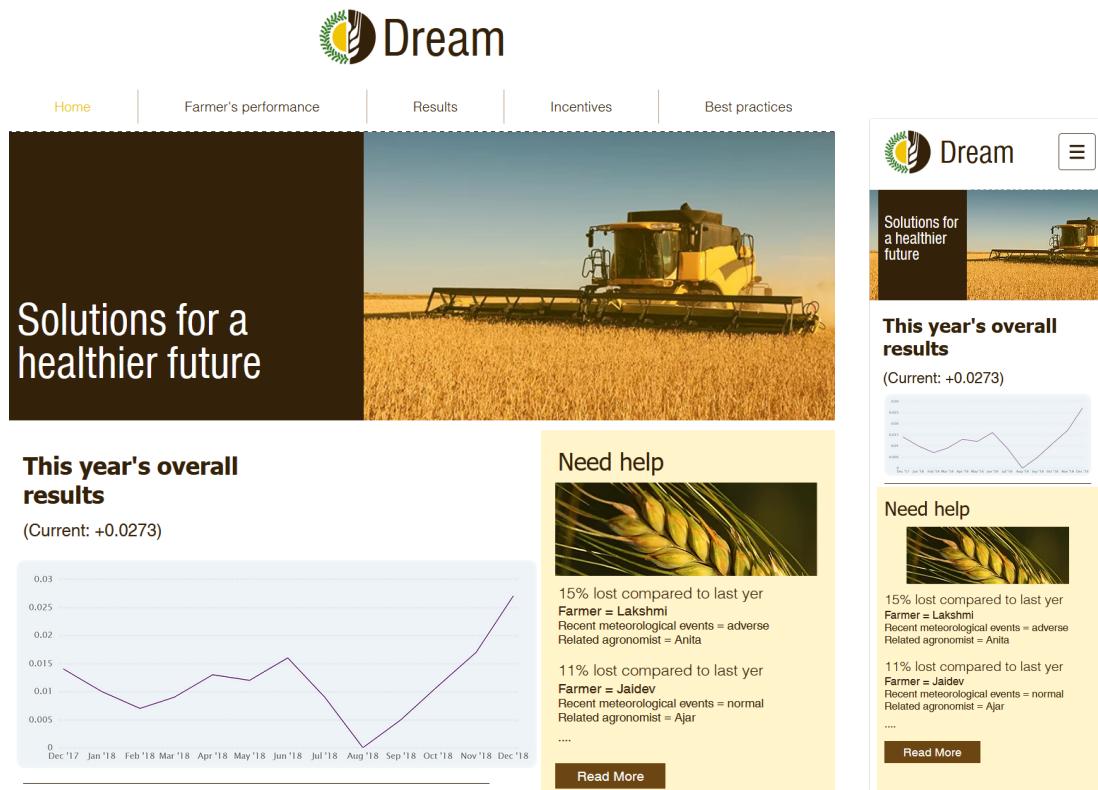


Figure 59: Policy Maker Home Page; web on the left, app on the right

- Policy Maker Visualize Farmers' performance

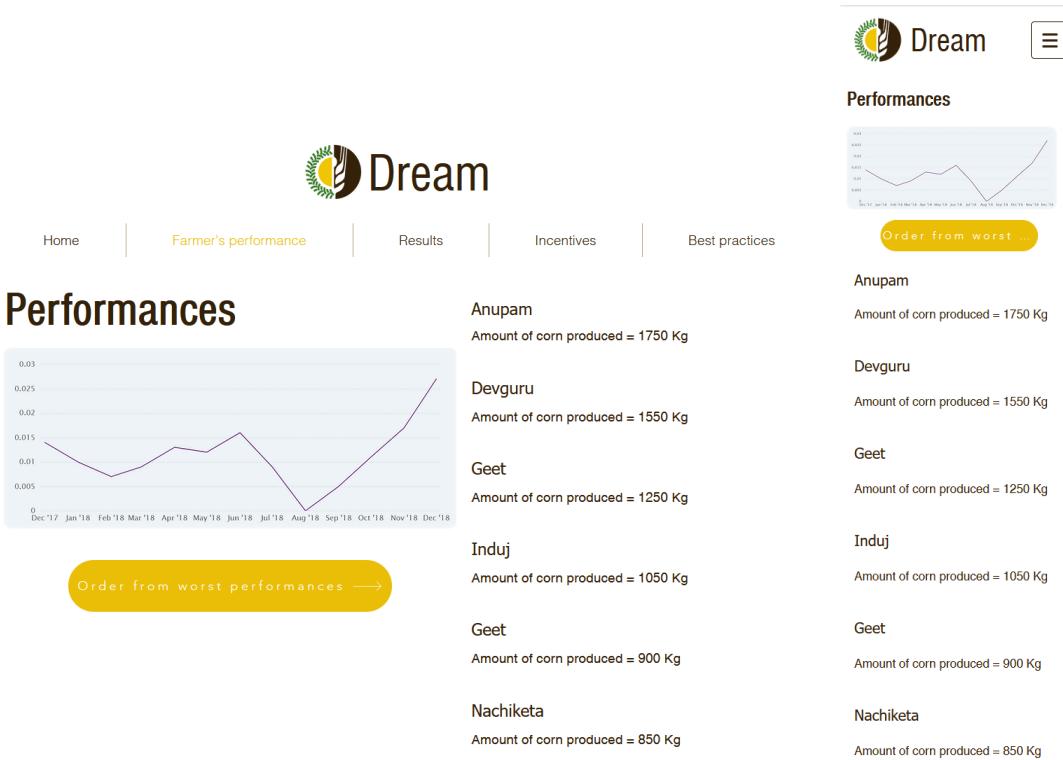


Figure 60: Policy Maker Visualize Farmers' Performance; web on the left, app on the right



Dream

- Home
- Farmer's performance
- Results
- Incentives
- Best practices

December 2021

Farmer: Anupam

Amount of corn produced = 1750 Kg
 Amount of energy used per kg = 2MJ
 Amount of fertilizer used per unit = 50 kg
 Amount of water used per unit = 2 625 000 l

[Ask for best practices →](#)



Dream

December 2021

Farmer: Anupam

Amount of corn produced = 1750 Kg
 Amount of energy used per kg = 2MJ
 Amount of fertilizer used per unit = 50 kg
 Amount of water used per unit = 2 625 000 l

[Ask for best practices](#)

Figure 61: Policy Maker Ask for Best Practice; web on the left, app on the right

- Policy Maker Best Practices

The image shows two side-by-side interfaces for the "Policy Maker Best Practices" application. On the left is the web version, and on the right is the mobile app version.

Web Interface (Left):

- Header: Dream
- Navigation: Home, Farmer's performance, Results, Incentives, Best practices
- Main Content Area:
 - Best-performing** (highlighted in brown)
 - Anupam: Amount of corn produced = 1750 Kg
 - Devguru: -Already asked- Amount of corn produced = 1550 Kg
 - Geet: Amount of corn produced = 1250 Kg
 - Induj: Amount of corn produced = 1050 Kg
 - Geet: -Already asked- Amount of corn produced = 900 Kg
 - Nachiketa: Amount of corn produced = 850 Kg

Mobile App Interface (Right):

- Header: Dream
- Icon: Person writing in a notebook
- Section: Best-performing
- Items:
 - Anupam: Amount of corn produced = 1750 Kg
 - Devguru: -Already asked- Amount of corn produced = 1550 Kg
 - Geet: Amount of corn produced = 1250 Kg
 - Induj: Amount of corn produced = 1050 Kg
 - Geet: -Already asked- Amount of corn produced = 900 Kg
 - Nachiketa: Amount of corn produced = 850 Kg

Figure 62: Policy Maker Best Practices Home Page; web on the left, app on the right

The screenshot shows the DREAM web application interface. At the top, there is a navigation bar with links: Home, Farmer's performance, Results, Incentives, and Best practices. The main content area has a brown header box containing the text "Best-practice request". Below this, the text "Request to: Anupam" is displayed. Underneath, there is a list of resource consumption data:

- Amount of corn produced = 1750 Kg
- Amount of energy used per kg = 2MJ
- Amount of fertilizer used per unit = 50 kg
- Amount of water used per unit = 2 625 000 l

Below the data, a message reads: "<<You have performed very well recently, can you share some advice?>>". A yellow "Send" button with a right-pointing arrow is located at the bottom of this section.

The screenshot shows the DREAM mobile application interface. It features a similar layout to the web version, with the "Best-practice request" header and the "Request to: Anupam" message. The resource consumption data is identical. The message "<<You have performed very well recently, can you share some advice?>>" is also present. A yellow "Send" button with a right-pointing arrow is at the bottom.

Figure 63: Policy Maker Ask for Best Practices through Best Practices; web on the left, app on the right

3.5 Flow of functionalities

3.5.1 User Sign Up and Login

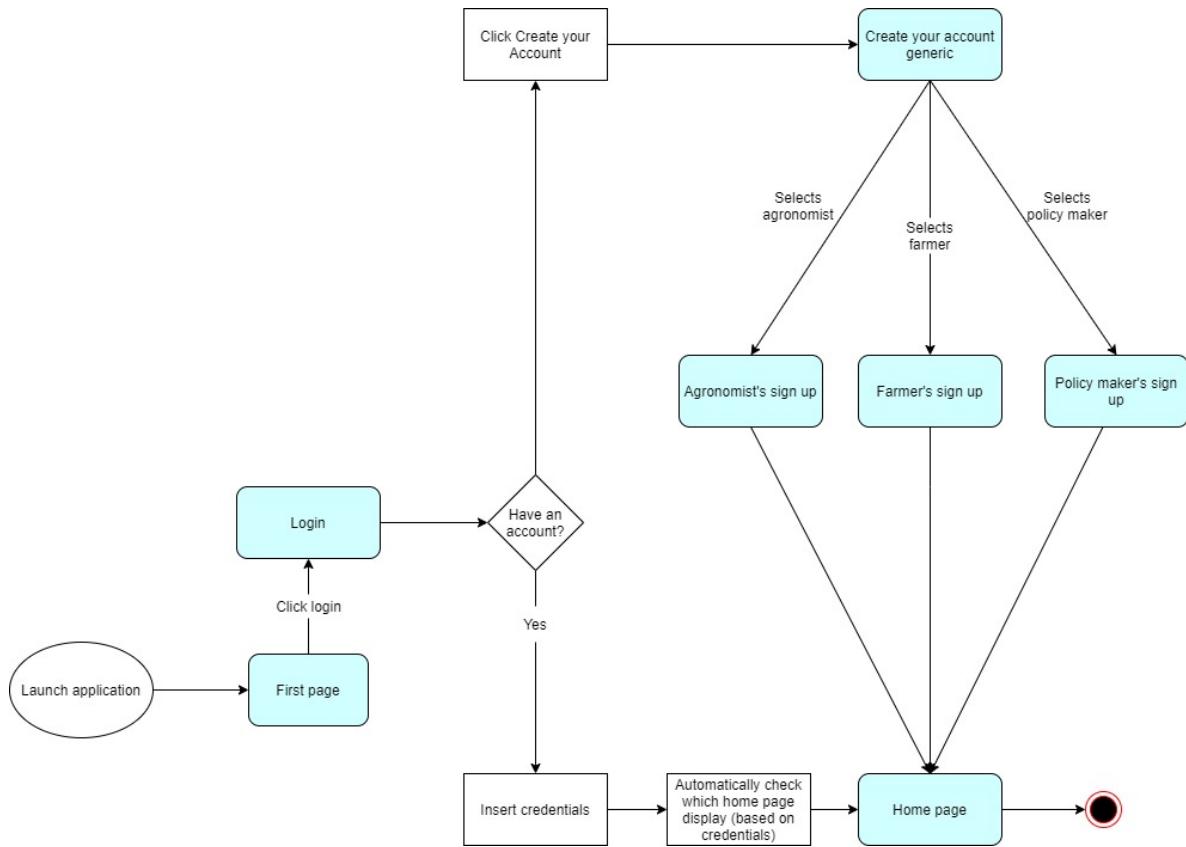


Figure 64: User Sign Up and Login

3.5.2 Farmer visualizes weather forecasts

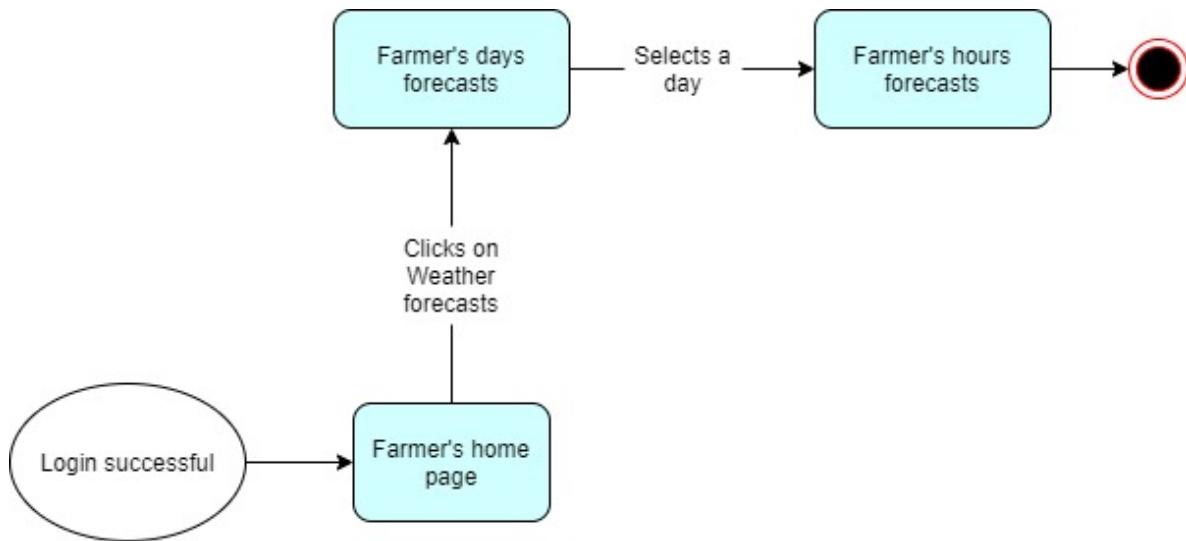


Figure 65: Farmer visualizes weather forecasts

3.5.3 Farmer visualizes personalized suggestions

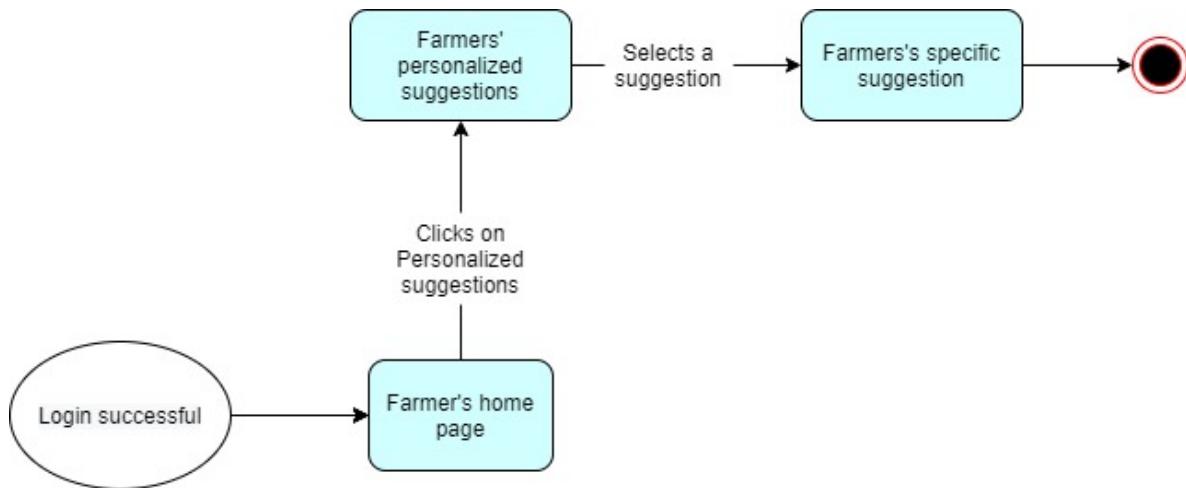


Figure 66: Farmer visualizes personalized suggestions

3.5.4 Farmer inserts production data

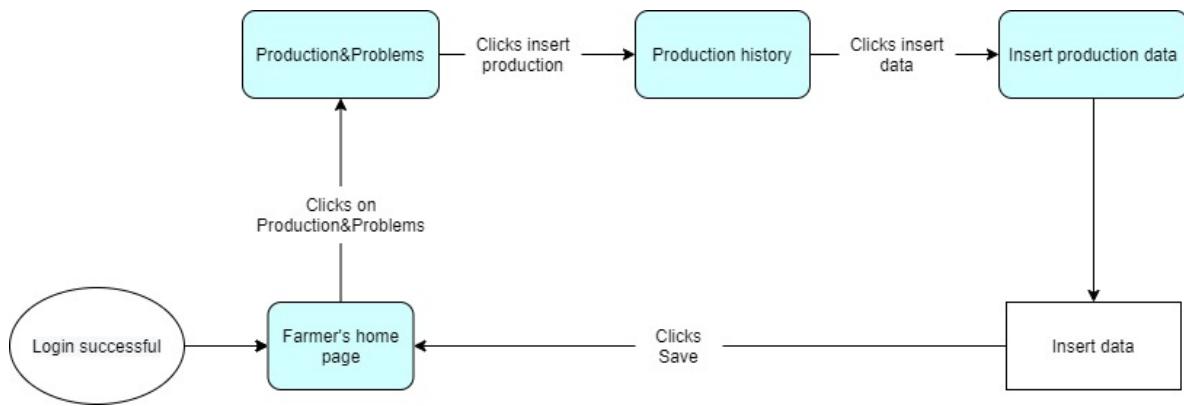


Figure 67: Farmer inserts production data

3.5.5 Farmer reports a problem

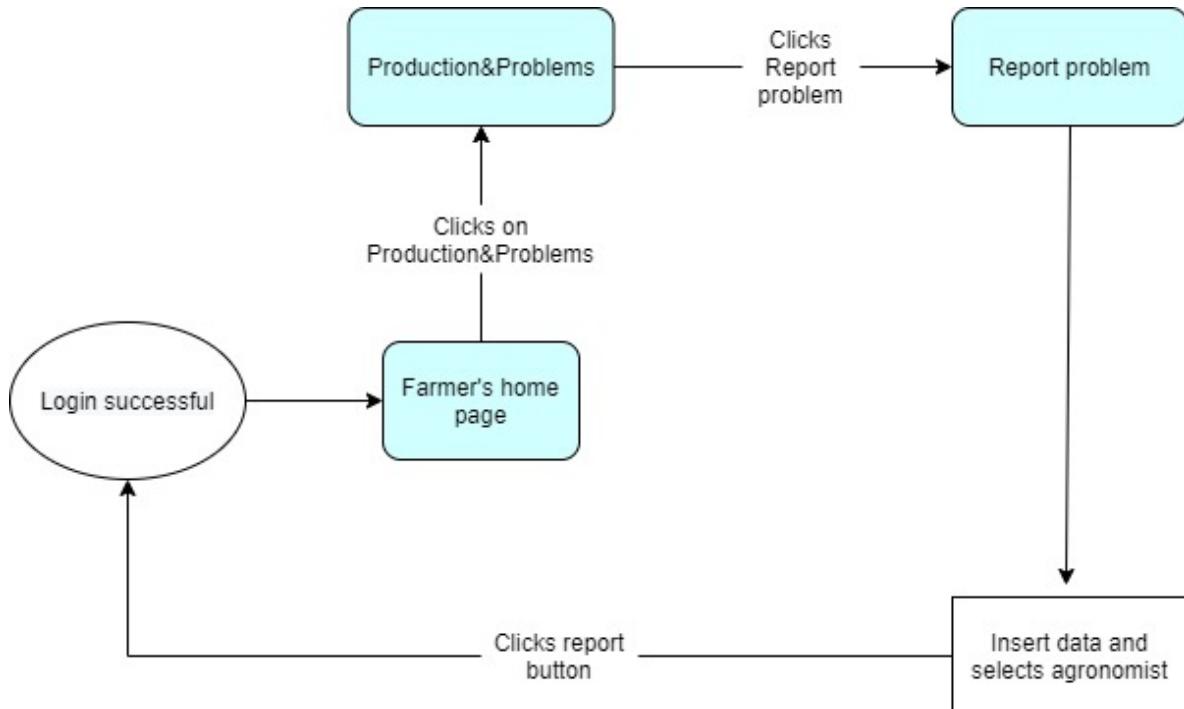


Figure 68: Farmer reports a problem

3.5.6 Farmer requests for help and suggestions

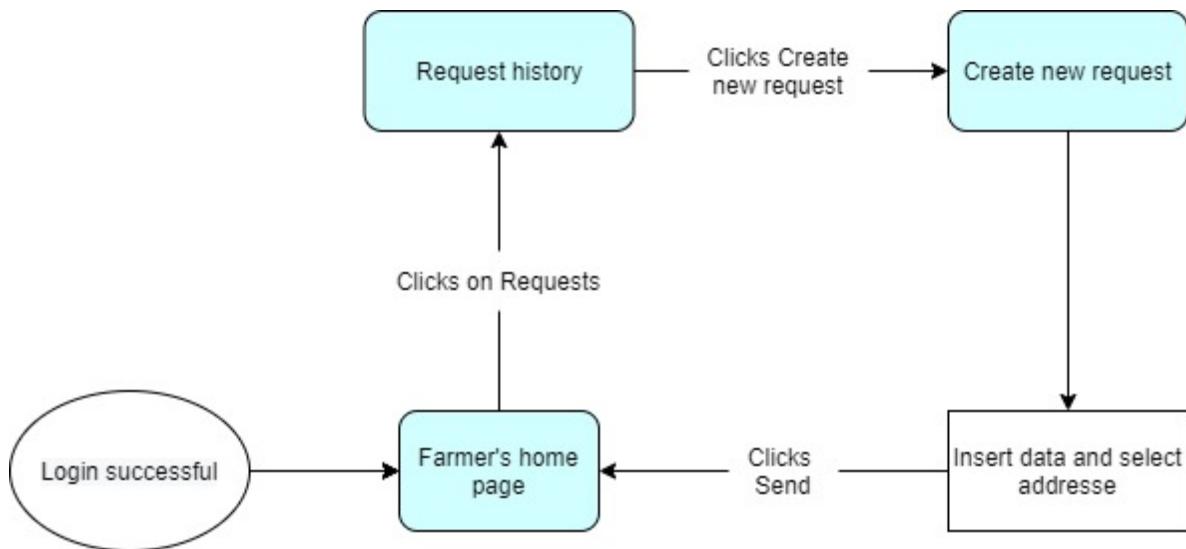


Figure 69: Farmer requests for help and suggestions

3.5.7 Farmer responds to a request for help and suggestions

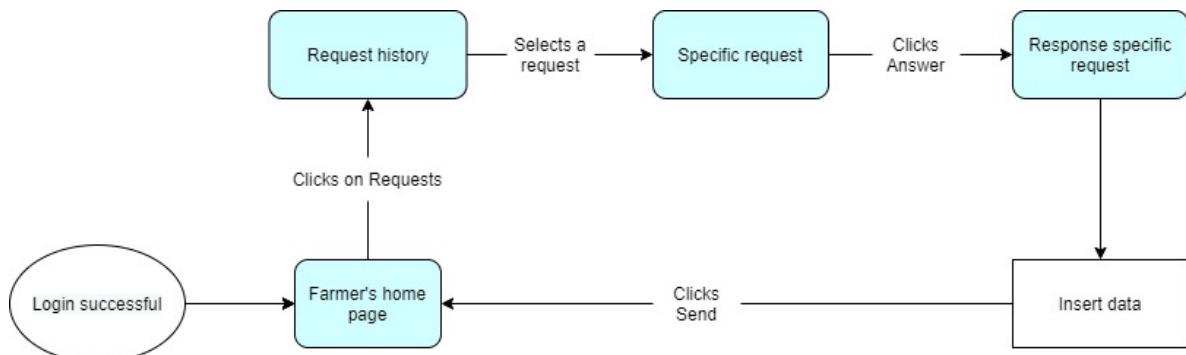


Figure 70: Farmer responds to a request for help and suggestions

3.5.8 Farmer writes a new post on forum

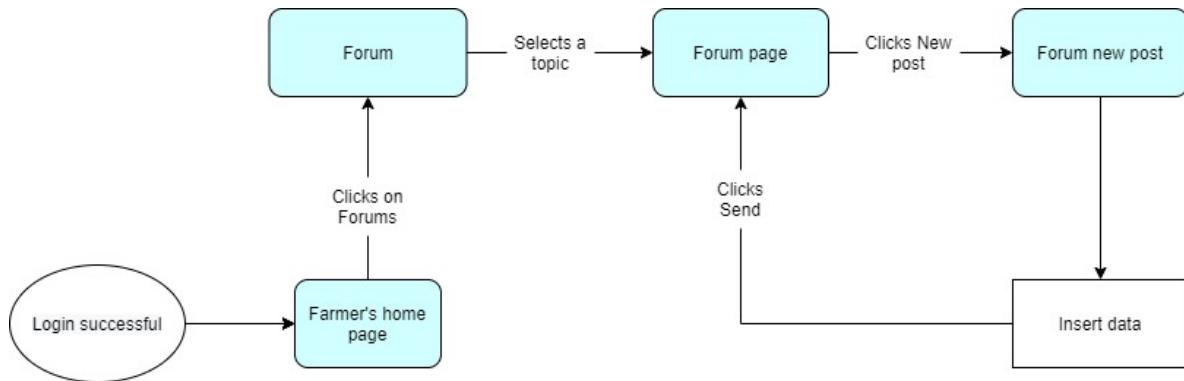


Figure 71: Farmer writes a new post on forum

3.5.9 Farmer responds on a discussion forum

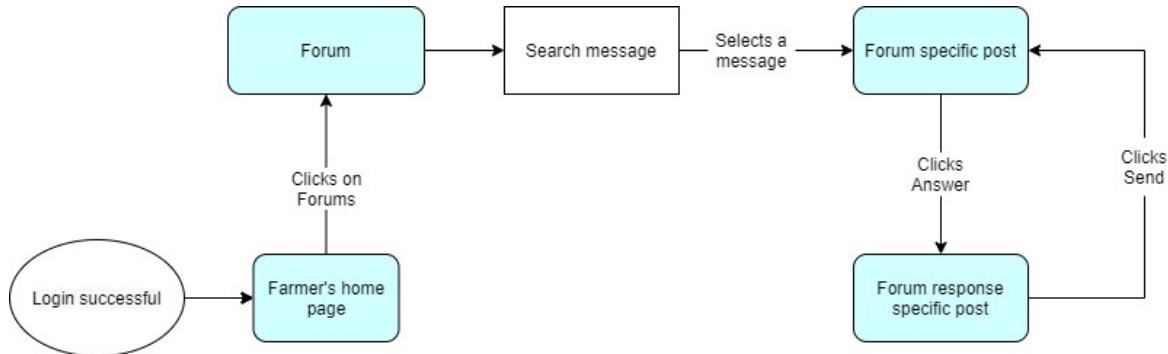


Figure 72: Farmer responds on a discussion forum

3.5.10 Farmer inserts best practices

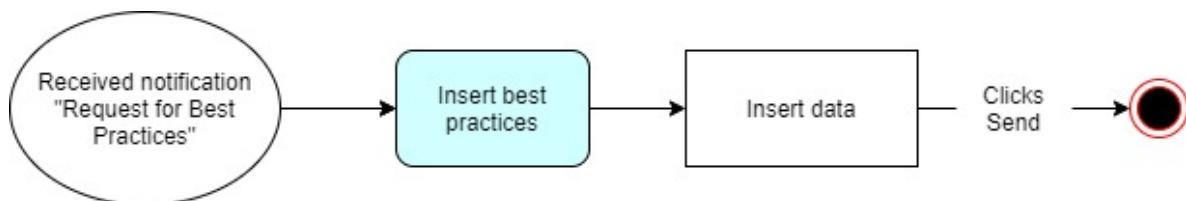


Figure 73: Farmer inserts best practices

3.5.11 Farmer subscribes to a topic

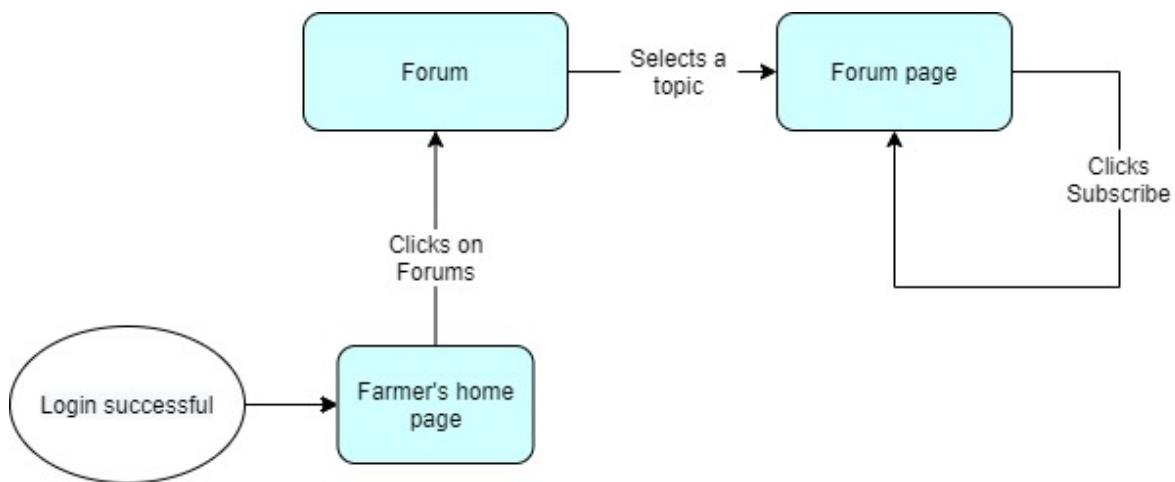


Figure 74: Farmer subscribes to a topic

3.5.12 Farmer unsubscribes from a topic

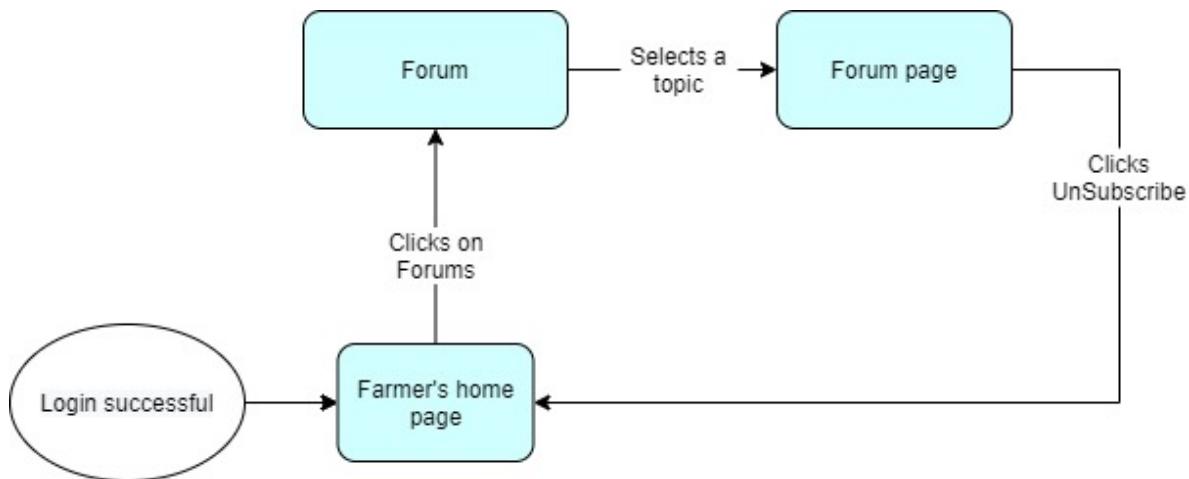


Figure 75: Farmer unsubscribes from a topic

3.5.13 Agronomist answers to a request for help and suggestions

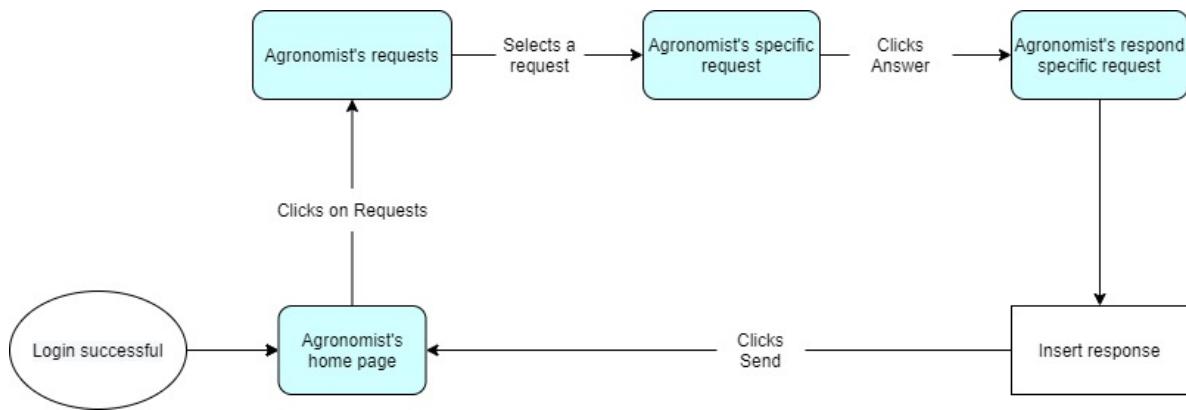


Figure 76: Agronomist answers to a request for help and suggestions

3.5.14 Agronomist visualizes daily plan

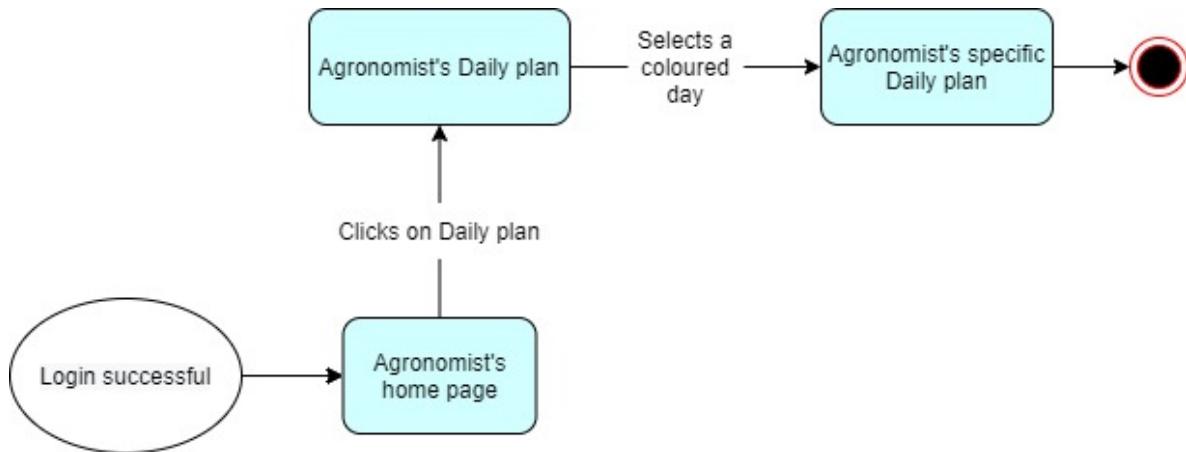


Figure 77: Agronomist visualizes daily plan

3.5.15 Agronomist inserts daily plan

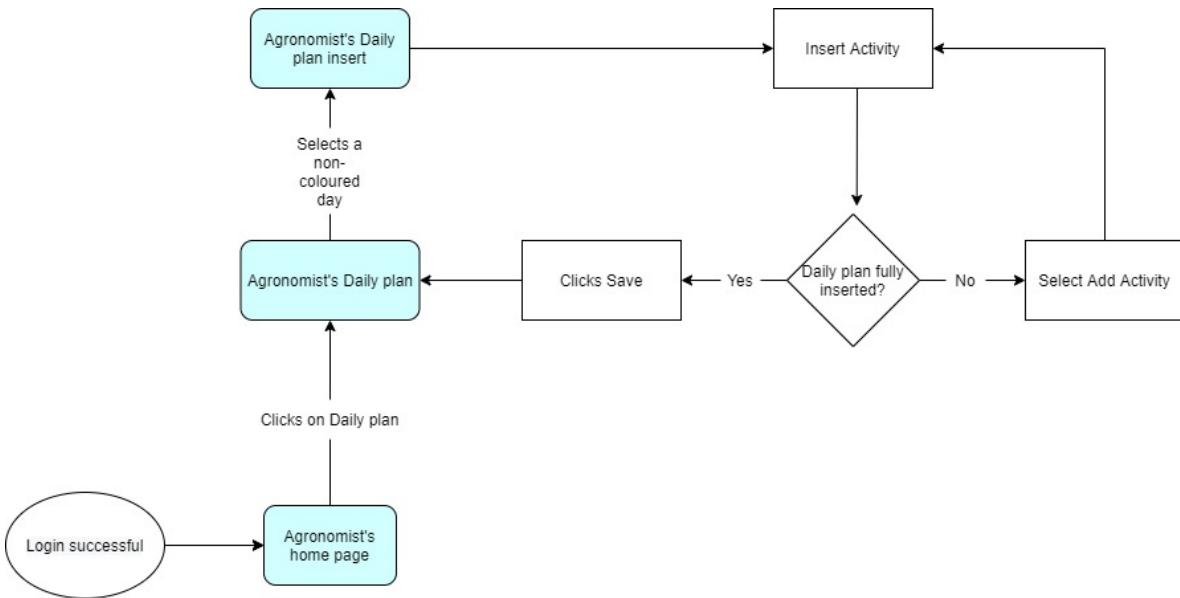


Figure 78: Agronomist inserts daily plan

3.5.16 Agronomist updates daily plan

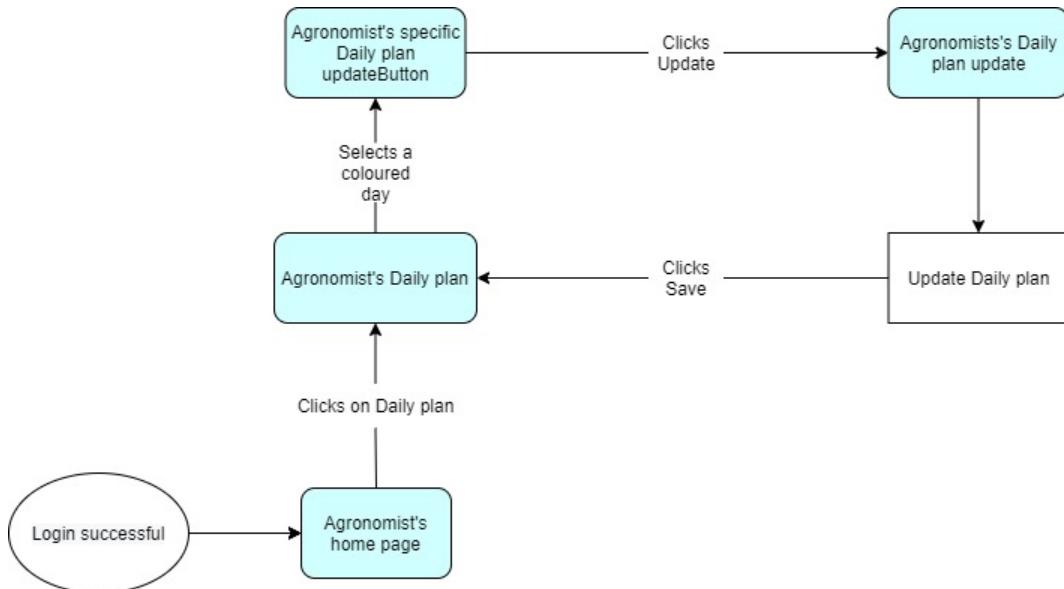


Figure 79: Agronomist updates daily plan

3.5.17 Agronomist confirms execution of daily plan, or specify deviations

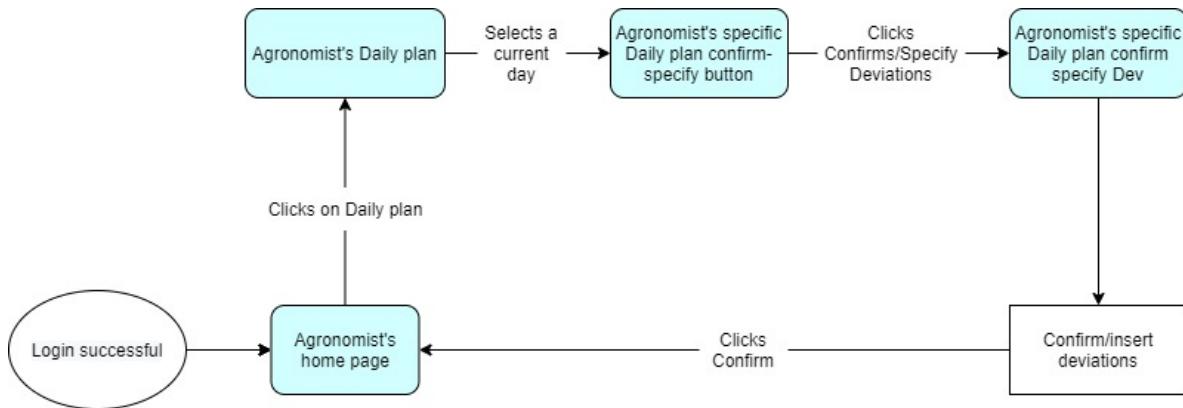


Figure 80: Agronomist confirms execution of daily plan, or specify deviations

3.5.18 Agronomist visualizes farmers' performance (production data)

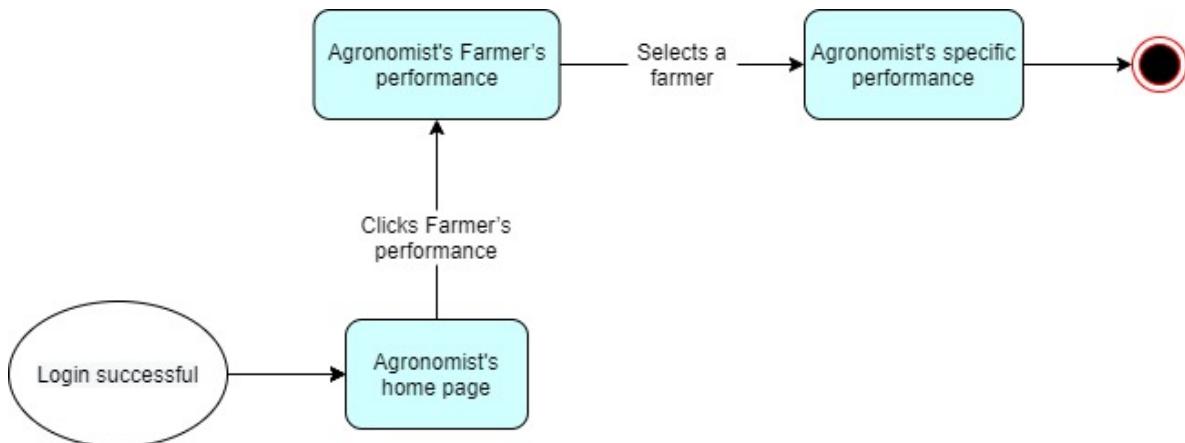


Figure 81: Agronomist visualizes farmers' performance

3.5.19 Policy Maker visualizes farmers' performance (production data)

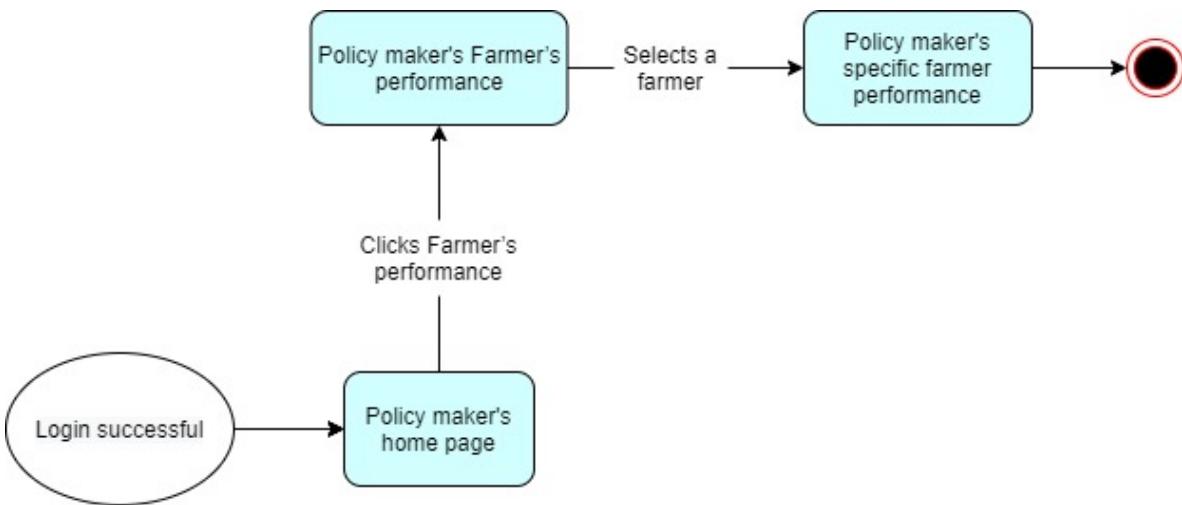


Figure 82: Policy Maker visualizes farmers' performance

3.5.20 Policy Maker asks for best practices to a farmer

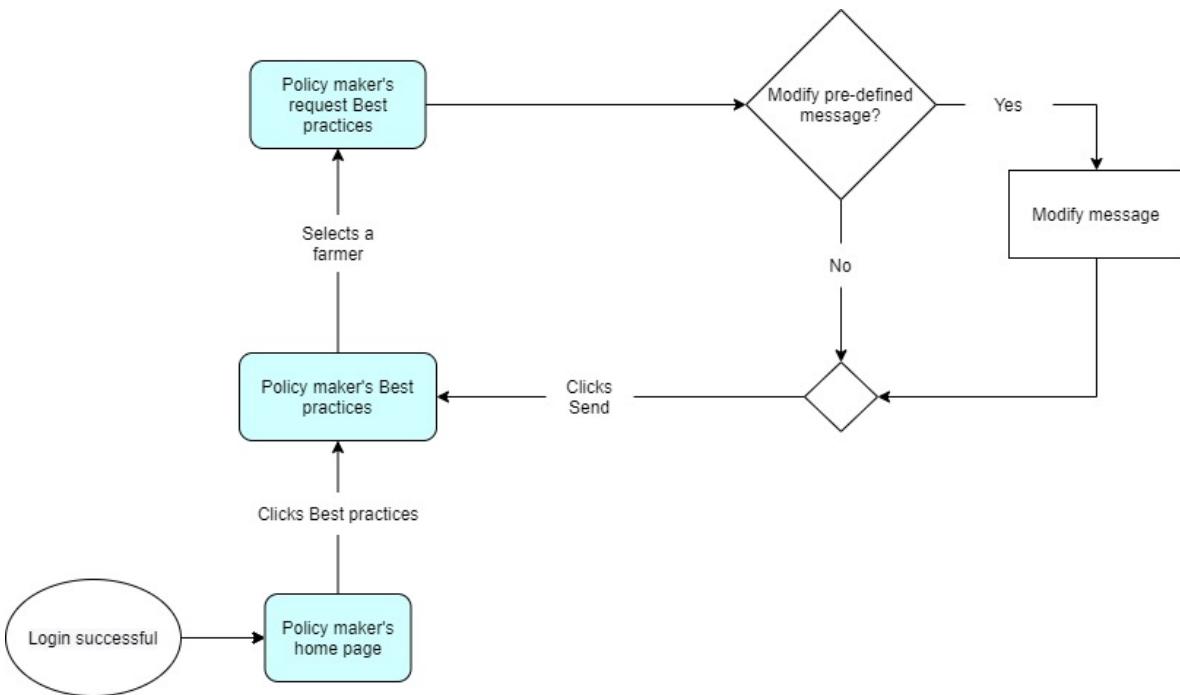


Figure 83: Policy Maker asks for best practices to a farmer

3.5.21 Policy Maker reports a bad performing farmer

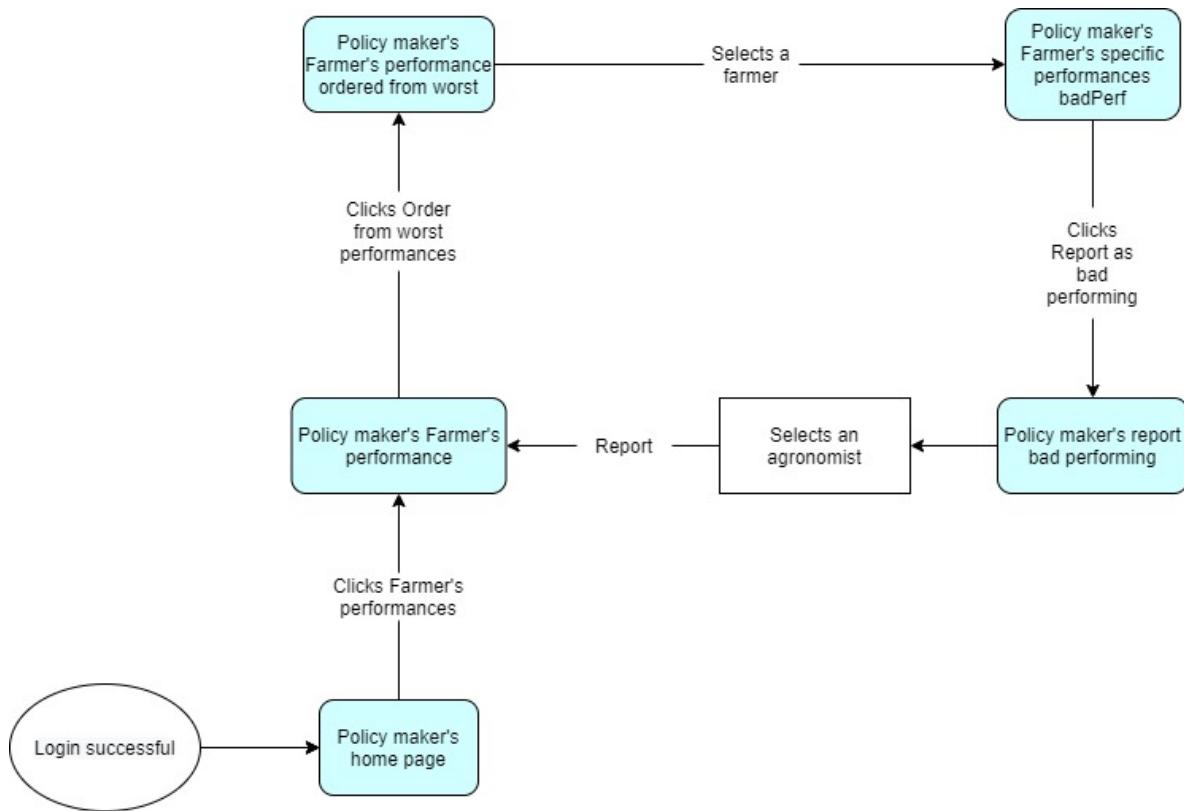


Figure 84: Policy Maker reports a bad performing farmer

4 Requirements Traceability

This section describes the traceability between requirements written in the RASD and modules described in the component diagram.

- **R1:** The system shall allow the registration of new users of three different types: farmer, agronomists and policy makers.
 - *ClientManager*: manages all guest user interaction, letting them signup and login.
- **R2:** The system shall elaborate a list of suggestions for the registered farmer user.
 - *Farmer Personalized Suggestions*
- **R3:** The system shall allow the farmer user to send help or suggestions requests to a selected agronomist user or farmer user.
 - *HelpAndSuggestion Controller*
- **R4:** The system shall send a notification to the farmer user when he receives requests for best practices or new answers or requests in the help requests or forum threads.
 - *EmailManager*: manages triggers that automatically send notification email to the chosen user.
- **R5:** The system shall allow a farmer user to read and respond to threads in the farmer forums.
 - *Forum Controller*
- **R6:** The system shall allow a farmer to insert production data.
 - *Performance data Controller*
- **R7:** The system shall allow the farmer user to respond to a request for best practices.
 - *Best practice Controller*
- **R8:** The system shall present an accurate weather forecast for the days following the current one to both farmer users and agronomists.
 - *Weather forecast Controller*
- **R9:** The systems shall allow an agronomist user to respond to help and suggestions requests received by farmer users.
 - *HelpAndSuggestion Controller*
- **R10:** The system shall send a notification to the agronomist user when a help request is received or a farmer is reported as “bad performing”.
 - *EmailManager*
- **R11:** The system shall present to agronomist users and policy maker users all the performance data that are available.
 - *Performance data Controller*
- **R12:** The system shall allow agronomist users to insert and modify a daily plan.
 - *DailyPlan Controller*
- **R13:** The system shall allow the agronomist to confirm the execution of the daily plan or specify its deviations.
 - *DailyPlan Controller*

- **R14:** The system shall not allow the creation of a daily plan if a farmer is not receiving it's mandatory visits, that must be at least twice a year, and the day is the last available. Consequently it shall notify the exception to the agronomist user.
 - *DailyPlan Controller*
 - *EmailManager*
- **R15:** The system shall present the daily plan to the agronomist user.
 - *DailyPlan Controller*
- **R16:** The system shall allow policy maker users to send to farmer users requests for best practices.
 - *Best practice Controller*
- **R17:** The system shall allow policy maker to report bad performing farmers to agronomists.
 - *Performance data Controller*
 - *EmailManager*

Also our design choice guarantees the system attributes defined in the RASD document, more precisely:

- *Reliability* and *availability*: the choice of stateless components, redundant servers and elastic load balancer guarantees that the system can adapt to any spike load and remain accessible at most times. Also there is no single point of failure, as if a component fails it can be rapidly repaired / substituted while other components absorb its load.
- *Security*: the use authentication token guarantees a user identity, the usage of HTTPS connections allow the system to encrypt all communication channels and data integrity is covered by the DBMS server which will perform data backups and by the stateless components which will not lose any data in case of failure.
- *Maintainability*: the use of the model-view-controller pattern is implemented and the stateless components give the system a high degree of separation, aiding the maintainability of the software base.
- *Portability*: the choice of both stateless components and thin clients guarantees a high level of portability of both clients and servers.

5 Implementation, Integration and Test Plan

Dream application can be divided as:

- *Client*: either browser or mobile
- *Web Server*
- *Application Server*
- *Internal Database*
- *External Services*: such as data concerning meteorological short-term and long-term forecasts

All the elements will be developed following a bottom-up approach in order to prevent the difficulty of creating complex stubs and the main focus will be on the application server.

In the following table we analyze all the functionalities presented in our RASD document and highlight the importance and the testing difficulty.

Functionality	Actor	Importance for customer	Difficulty
Login	General User	High	Low
SignUp	General User	High	Low
Visualize relevant data	Farmer	High	Medium
Insert data about production and problems in the system	Farmer	Medium	Low
Request and response for help and suggestions	Farmer	High	High
Participate in a discussion forum	Farmer	Medium	High
Receive and answer to farmers requests for help	Agronomist	High	High
Visualize weather forecasts	Agronomist	Medium	Low
Visualize, insert, and update daily plan	Agronomist	High	Medium
Confirm the execution of the daily plan	Agronomist	Medium	Low
Specify deviations from the daily plan	Agronomist	Medium	Low
See information about crops and farmers	Policy Maker	High	Medium
Select well performing farmers and ask best practices	Policy Maker	Medium	High
Select bad performing farmers and send suggestions	Policy Maker	Medium	High

Before starting the test phase it is important to underline that all the functionalities rely on the DBMS of the internal database, so surely it needs to be implemented and tested as the first component, followed by the TokenManager, the EmailManager and an integration test.

Then it would be a good idea to proceed by difficulty, from the more difficult components to the easier ones.

Another strategy could also be to implement and test all the functionalities of an actor (farmer/agronomist/policy maker) before proceeding to the following one.

We will follow a hybrid approach, starting with all general farmer's functionalities, followed by agronomist ones and then policy maker ones, but still proceeding by difficulty for each actor.

This will lead to the possibility of releasing as soon as possible a first version of the software to farmers users and to implement and test agronomists and policy makers' functionalities while farmers start to use their own ones (at the beginning, when no farmer has inserted his production, nobody has requested for help, etc... policy makers and agronomists would probably have little to do, so it is possible to develop their functionalities while the forum gets filled and productions start to get inserted).

- *Login - SingUp:* Since in order to perform a login it is necessary to signUp before, the login functionality needs to be implemented after the signUp. Considering also that the signUp and login phase are strictly related, it would be a good idea to implement the 2 functionalities together and then proceed with a unit of the Client manager, followed by an integration test with the TokenManager and the EmailManager components. Since the 2 functionalities also require interactions with the DBMS (i.e. to retrieve an already registered account) an integration test with the DBMS needs to be performed too.
- *Visualize relevant data:* For this functionality it is required to implement and test the Weather Forecast Controller. This requires the external API which provides the weather forecasts, so an integration test needs to be performed in order to check the communication. In addition, in order to provide the personalized suggestions functionality, Farmer Personalized Suggestion component needs to be implemented and tested, together with an integration test with the DBMS (to retrieve the stored suggestions).
- *Insert data about production and problems in the system:* For this functionality it is required to partially implement and test the Performance Data Controller. This component also provides the access to farmers' productions to agronomists and policy makers, but at first only the farmer functionality will be developed. An automated unit test will be performed, followed by an integration test with the DBMS (since productions get stored in the database).
- *Request and response for help and suggestions:* For this functionality it is required to implement and test the HelpAndSuggestion Controller component. A unit test will be performed (i.e. to check the correct delivery and notification of messages to the correct addressee) followed by an integration test with the DBMS (required to retrieve the necessary messages stored in the database) and the EmailManager (required to send notifications).
- *Participate in a discussion forum:* For this functionality it is required to implement and test the Forum Controller component. A unit test and an integration test with the EmailManager will be performed in order to check the correct delivery and notification(topic subscription) of messages, followed by an integration test with the DBMS required to retrieve all the messages exchanged on the forum (stored in the database).
- *Receive and answer to farmers requests for help:* This functionality requires the HelpAndSuggestion Controller component already implemented for the analogous farmer functionality. Since an agronomist is only allowed to answer to requests, it is enough to implement this constraint and perform again the unit and integration test of the component, with just this new control added in the unit test.

- *Visualize, insert, and update daily plan*: For this functionality it is enough to extend the Weather Forecast Controller, already implemented and tested for farmers' analogous functionality, in order to grant the functionality also to agronomists. A unit test will follow.
- *Visualize, insert, and update daily plan - Confirm the execution of daily plan - Specify deviations from daily plan*: Since all the functionalities are strictly related, uses the DailyPlan Controller component and the Confirm/Specify Deviations are simple to test, all 3 functionalities can be implemented together. A unit test (used to check the correct update of plans, the correct confirmation/specification of deviations,...) will then be performed, followed by an integration test with the DBMS (required since the database stores the daily plans).
- *See information about crops and farmers*: This functionality requires the Performance Data Controller component, already implemented for the "Insert data about production and problems in the system" farmer's functionality. The component will be extended with the possibility for the policy maker to check all the productions inserted. A unit test will then check that the previous function offered by the component continues to work correctly (through the automated test already created) and verify for the policy makers the possibility to visualize all the productions inserted. An integration test with the DBMS will follow.
- *Select well performing farmers and ask them best practices*: For this functionality it is required to implement and test the Best practice Controller component, then used together with the Performance Data Controller component in order to retrieve the correct performances of farmers. A unit test will be performed followed by an integration test with the Performance Data Controller, EmailManager and DBMS (in order to check the correct creation, delivery, notification, ...).
- *Select bad performing farmers and send them suggestions*: This functionality requires the Performance Data Controller component, in order to visualize farmers' performances and the Email Manager in order to report bad performing farmers to agronomists and take advantage of their suggestions, both already implemented. It is enough to extend the Performance Data Controller component to support this functionality and proceed with a unit test (to check also previous functionalities offered by the component) and an integration test with the EmailManager and the DBMS.

5.1 Clarification on component integration

In this chapter we will describe graphically how components will be integrated, following what explained in previous paragraphs.

5.1.1 Integration of the components of the ApplicationServer that exploits the DBMS

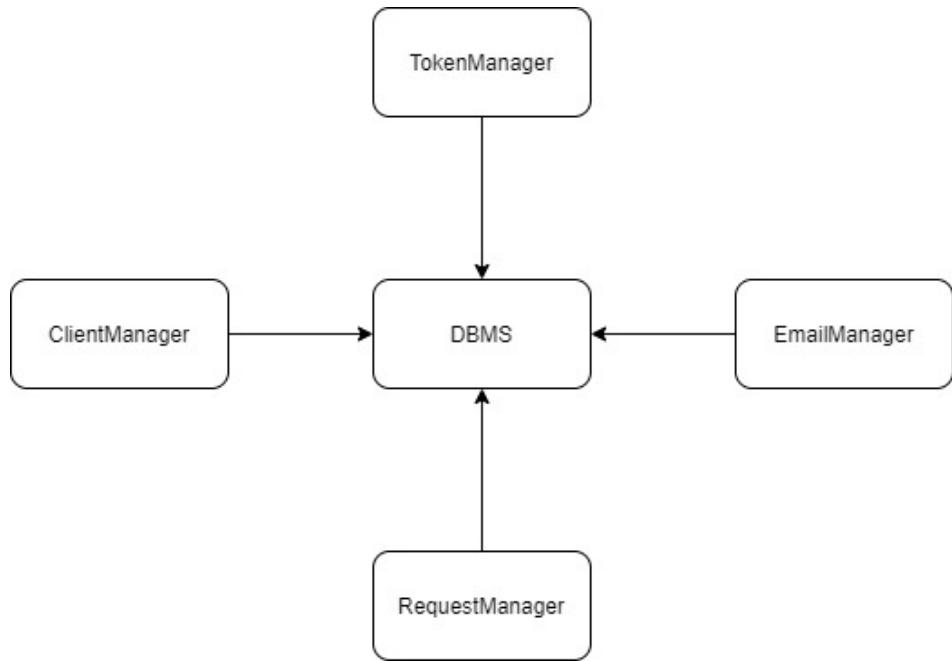


Figure 85: Components Integration

5.1.2 Integration with external services

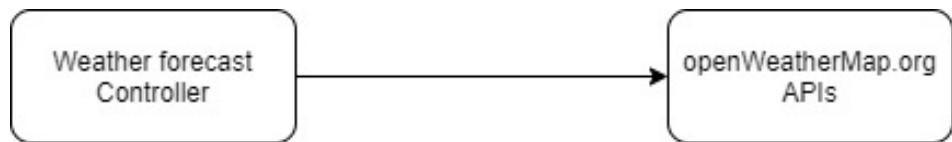


Figure 86: External APIs integrations

5.1.3 Integration of RequestManager and ClientManager

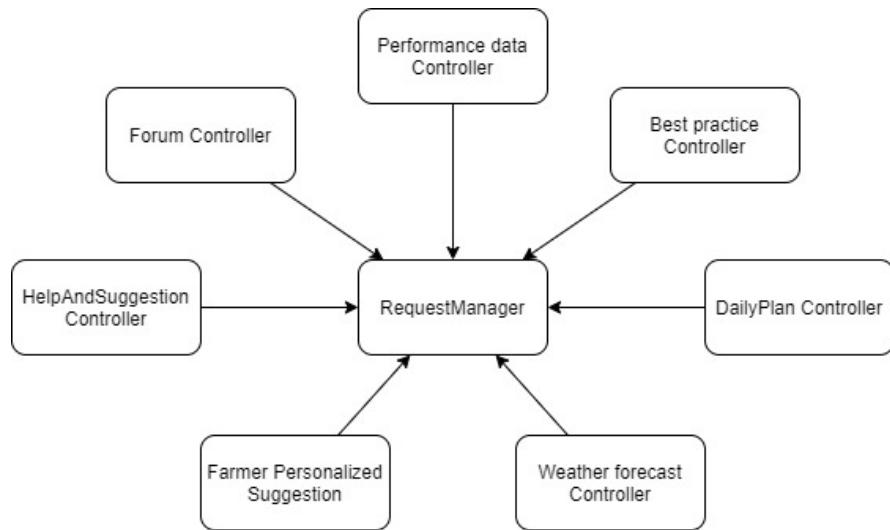


Figure 87: Request Manager Integration

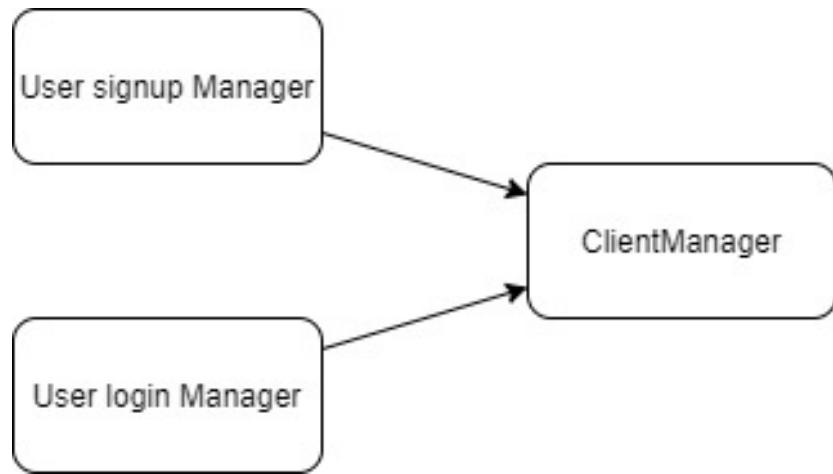


Figure 88: Client Manager Integration

5.1.4 Integration of Web Browser module and Mobile Application module with the Web Server

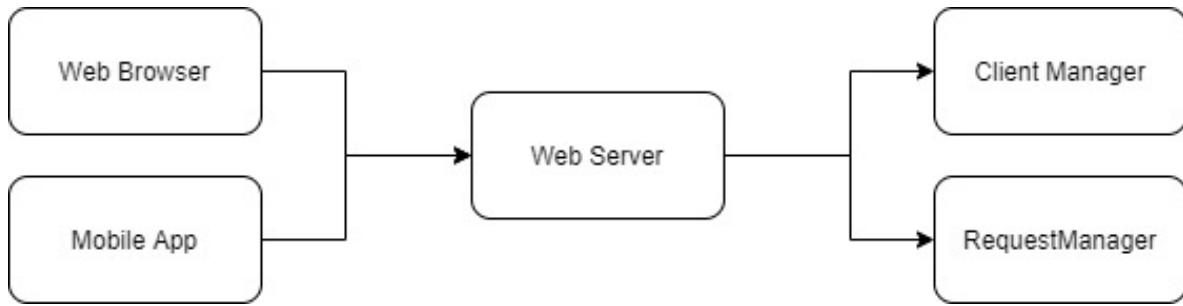


Figure 89: Client-Server Integration

6 Effort Spent

	Cap1	Cap2	Cap3	Cap4	Cap5	Latex
Brunati	1h	1h	18h	1h	5h	
Cappelletti	2h	9h	2h	1h	1h	8h
Curti	1h	12h	1h	3h	1h	