



POLITECNICO
MILANO 1863

Implementation and Test deliverable

Authors: Pietro Valente
Andrea Seghetto

Professor: Damian Andrew Tamburri

Version: 1.0

Date: 6/2/2022

Repository GitHub: <https://github.com/pietrovalente/DREAM-software-engineering-2>

Copyright: Copyright © 2021, Pietro Valente & Andrea Seghetto – All rights reserved

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Definitions, Acronyms	2
1.3.1	Definitions	2
1.3.2	Acronyms	3
1.4	Revision history	3
1.5	Document Structure	3
2	Functions implemented	4
2.1	Sign up and log in	4
2.2	Homepage	9
2.3	Navbar	10
2.4	View/update farm data	11
2.4.1	Old crops	13
2.4.2	View day data	14
2.4.3	Modify/update data	15
2.4.4	Update/delete land	16
2.5	Ask for help	17
2.6	View ranking	18
2.7	Report/advise	19
2.8	Forum	21
2.9	Weather forecast	23
2.10	Log out	24
3	Prototype limitations	25
3.1	Security	25
3.2	Score	25
3.3	Forum	25
3.4	Ranking	25
3.5	Land daily update	25
4	Development frameworks adopted	26
5	Source code structure	27
6	Test cases	29
6.1	Access tests	29
6.1.1	LogInServlet test suite	29
6.1.2	LogoutServlet test suite	30
6.1.3	SignUptServlet test suite	30
6.1.4	SignUpFarmServlet test suite	33
6.2	Home tests	33
6.2.1	HomeServlet test suite	33
6.3	Farm tests	34
6.3.1	ViewDataServlet test suite	34
6.3.2	UpdateDataServlet test suite	34
6.3.3	AddLandServlet test suite	34
6.3.4	DeleteLandServlet test suite	35

6.4	Ask for help tests	35
6.4.1	AskForHelpServlet test suite	35
6.5	Ranking tests	36
6.5.1	ViewRankingServlet test suite	36
6.6	Report tests	36
6.6.1	ReportAdviseServlet test suite	36
6.7	Forum tests	37
6.7.1	ForumServlet test suite	37
6.7.2	DiscussionDetailsServlet test suite	38
6.8	Weather forecast tests	38
6.8.1	WeatherForecastServlet test suite	38
7	Installation instructions	39
8	Effort Spent	40

1 Introduction

1.1 Purpose

The purpose of this document is to provide more technical and detailed information about the implementation and testing of the system previously presented in the RASD and DD documents.

It will represent a strong guide for the users as well as for everyone wanting to test the functionalities implemented. It is therefore a document in which the implementation choices made, the system functions offered and the frameworks used for their development will be exposed.

The structure of the source code will be analyzed and the set of tests performed will be reported and described. The installation instructions will also be listed, i.e. the prerequisites for using the system with everything necessary and useful to install and run the software. In general, the main different features listed in this document are:

- Requirements/functions implemented in the software
- Adopted development frameworks
- The structure of the source code
- Information on how testing was performed
- The installation instructions

Stakeholders are invited to read this document in order to understand the features of the system being aware of the choices that have been made to offer all the functionalities satisfying the established functional and non-functional requirements.

1.2 Scope

We will deal with the implementation and testing of DREAM, which is a data-driven predictive and also distributed system designed for governmental use and realized through the collaboration between the government of Telengana and some IT providers which must interface with multiple stakeholders.

The main goal of the whole system is to monitor and optimize the farming production of the Indian state Telengana on which a large part of the population depend, making it resilient to the many difficulties that threaten it, including phenomena related to climate change.

As a distributed system we foresee the existence of a central server that include the front-end part, that is a user interface for webapp and application, with a simple design, to cope with the poor digital education of farmers and also back-end part that is a Database Management System (DBMS) which will take care of the storage and management of data.

The system will use some external tools/APIs (i.e. TSDPS, Google Maps) to make available to its users some important features. One of these features lies in showing predictions about short or long-term climatic events in order to anticipate and govern them in the best possible way.

The system is supposed to have three types of stakeholders: farmers, agronomists and policy makers. For each of these figures, the system provides some functions that will be described below. As for policy maker the system guarantees them the ability to monitor the evolution of the entire national production process, distinguishing the farmers who have been more successful from those who needs help and allowing them to analyze whether the advice given by agronomists leads to significant results.

To identify the best farmers, an annual ranking of the farms registered is drawn up in real time until the deadline. In such a ranking each farm has its own score computed by the system following a precise formula and only the ten farms with the best score will be awarded; if a tie between them occurs all farms will be rewarded.

Policy makers have the opportunity to see the ranking status with the efficiency scores of the farms in real time and they reward the best farms only once a year, then the ranking is reset.

As for agronomists the system must assure them to write reports on the farms they visit containing the advice they have given with the help provided by DREAM AI; namely an AI algorithm of the system which proposes possible general suggestions on the production of each farm that the agronomist can choose to use or not. Only agronomists are allowed to give advice because they have the skills to do so, DREAM AI is just a tool used to help agronomists improving their advice which works by analyzing the evolution of the farm's production overtime and comparing it to the data related to other farms. The agronomists selected by the government are trained to give effective advice against climate change. Each agronomist is associated with an area, which corresponds to a province of Telengana.

Finally with regard to farmers, the system must allow them to view relevant information such as weather predictions and to request the intervention of the agronomist associated with them in case of in difficulty then they will take advantage of the personalized suggestions provided by him after the visit to the farm to improve production.

Furthermore, it will be possible for each farmer to interface with others by asking them for help or by creating a new discussion on the related forum. The advice given in this area remains informal and its effectiveness is not monitored.

The few paragraphs just read represent an overview of the main functionalities offered by the whole system, the prototype of DREAM developed which we are going to present deals only with the functionalities affecting farmers, simulating the acquisition of data from the other parts of the system.

Between the two methods of access and use to the system envisaged, in this prototype we will consider only access via webapp, reserving the implementation of the smartphone application for future developments. In order to show the main services offered by DREAM to registered farmers, some simplifications which will be better described afterwards have been introduced, for example in the computation of the score or in the updating of the ranking.

The reading of this document should be preceded and accompanied by the reading of the RASD and the DD documents, in order to better understand the functioning of the developed software.

1.3 Definitions, Acronyms

1.3.1 Definitions

Validation code	These codes are used to identify users for who they really are. Each user will receive an email invitation to register on the platform, with a personal code for registration. Validation verifies the correctness of the code.
Validation data	Each time data is entered by the user, a validation process verifies that the characters do not exceed the maximum number and that there are no harmful strings (i.e. SQL injection).
Query	Request made to the system in order to find information.
Webapp	Application software that runs on a web server.
Zone	A province of Telengana to which an agronomist is associated.
JUnit5	Unit testing framework for the Java programming language.

1.3.2 Acronyms

RASD	Requirements Analysis and Specification Document
DREAM	Data-dRiven PrEdictive FArMing in Telengana
DREAM AI	Data-dRiven PrEdictive FArMing in Telengana Artificial intelligence
TSDPS	Telangana State Development Planning Society
API	Application Programming Interface
DD	Design Document
SQL	Structured Query Language
JEE	Java Enterprise Edition

1.4 Revision history

For the history of the creation and modification of the document refer to the link: <https://github.com/pietrovalente/ValenteSeghetto/commits/main>

1.5 Document Structure

- **Chapter 1** describes the scope and purpose of the document, including its structure and the set of definitions, acronyms and abbreviations used.
- **Chapter 2** aims to provide a description of the functionalities implemented; it is the core section of the document. More precisely, this section describes which goals and requirements have been chosen and how they have been developed.
- **Chapter 3** provides a thorough examination of the development framework adopted.
- **Chapter 4** contains the description and the analysis of the structure given to the source code.
- **Chapter 5** includes all the information about the test carried out; namely which test cases have been considered, how they have been performed and which outcome they produced.
- **Chapter 6** is devoted to installation instructions. Here system users and testers will find the information useful to install and run the software.
- **Chapter 7** shows the effort spent for each member of the group.

Finally, there is a section dedicated to all the references used.

2 Functions implemented

The prototype of DREAM created was developed as an Eclipse JEE Dynamic Web Project with the features that will be exposed in the next sections, aimed at reproducing all the main services provided for farmers connected to the system via WebApp. Among the DREAM functions proposed in the RASD and in the DD it was therefore decided to discard all those not aimed at farmers and instead develop the following dedicated to them.

2.1 Sign up and log in

The first implemented functions offered to users are registration and log in to the system. These functions contribute to the achievement of the following goal and the implementation of the following requirements:

G10	The system must distinguish users in 3 categories: farmers, policy makers and agronomists
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
D12	The user picks up the right category
R35	The system must show a specific different form to compile to each user, based on the category he selects, during registration
R36	The system must ask each user to fill in the form to complete the registration
R37	Each user can view only the features specific to his category

Both services are displayed on the initial screen which is reached after running the software; with regard to registration, after clicking the appropriate sign-up item, the user is shown a screen in which he is first required to enter a code.

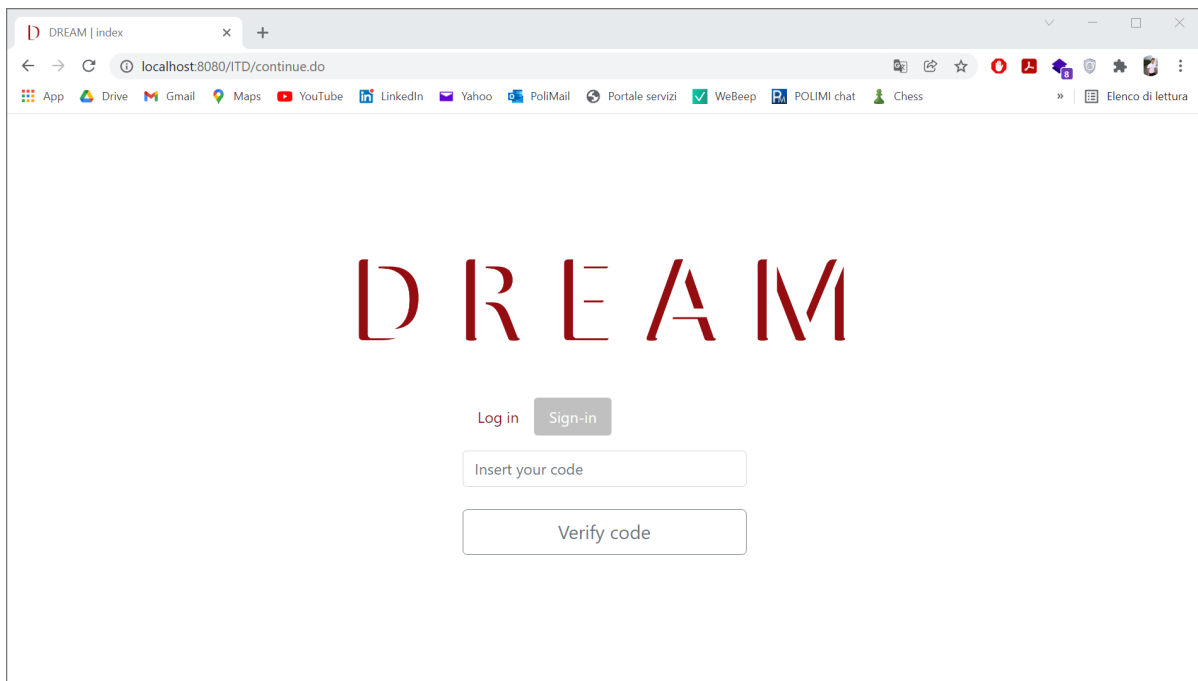
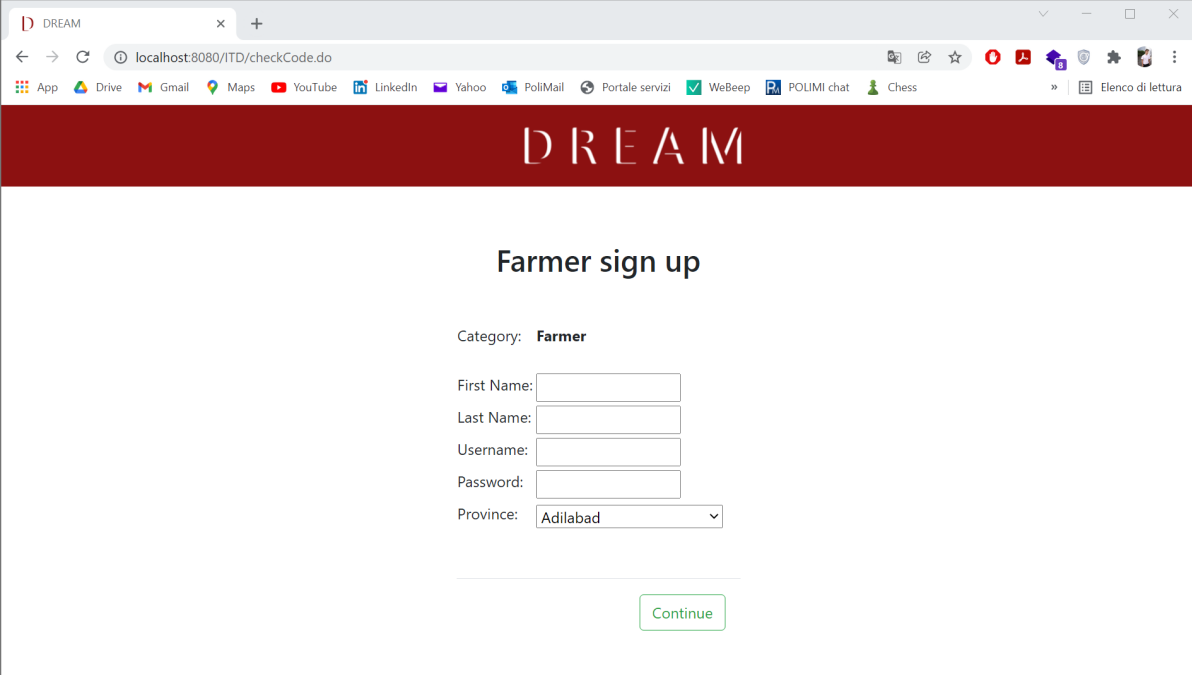


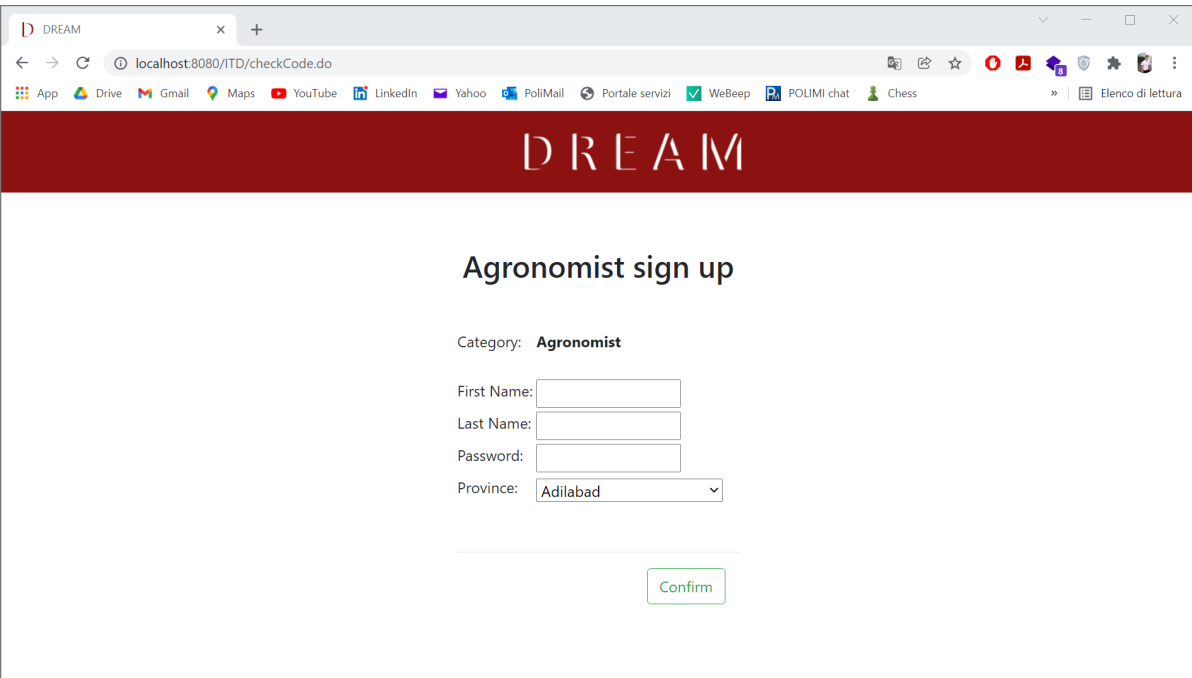
Figure 1: Check code

It is a government code delivered to the farmer by the government, which is necessary in order to proceed with the actual registration in the system. The code is useful for the government to manage the influx of new users to the platform and thus avoid the registration of unexpected users namely ones that do not fall into one of the three categories of users mentioned above: farmers, agronomists and policy makers. If you try to enter an incorrect code, an alert message reporting "Code not valid" will be displayed if, on the other hand, you enter a correct code then you are directed to a new screen and the code entered will no longer be available / usable because it has been consumed.



The screenshot shows a web browser window with the URL `localhost:8080/ITD/checkCode.do`. The page has a dark red header with the word "DREAM" in white. Below the header, the title "Farmer sign up" is centered. The form includes a "Category:" label with the value "Farmer". There are input fields for "First Name:", "Last Name:", "Username:", and "Password:". Below these is a "Province:" label with a dropdown menu showing "Adilabad". A green "Continue" button is at the bottom right of the form area.

Figure 2: Farmer Sign Up



The screenshot shows a web browser window with the URL `localhost:8080/ITD/checkCode.do`. The page has a dark red header with the word "DREAM" in white. Below the header, the title "Agronomist sign up" is centered. The form includes a "Category:" label with the value "Agronomist". There are input fields for "First Name:", "Last Name:", and "Password:". Below these is a "Province:" label with a dropdown menu showing "Adilabad". A green "Confirm" button is at the bottom right of the form area.

Figure 3: Agronomist Sign Up

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ITD/checkCode.do'. The browser's toolbar includes various icons for applications like Drive, Gmail, Maps, YouTube, LinkedIn, Yahoo, PolMail, Portale servizi, WeBeep, POLIMI chat, and Chess. The page has a dark red header with the word 'DREAM' in white capital letters. Below the header, the main content area is white and contains the title 'Policy Maker sign up' in a bold, black font. Underneath the title, the text 'Category: Policy Maker' is displayed. The registration form consists of three input fields: 'First Name:', 'Last Name:', and 'Password:', each followed by a white rectangular text box. A green 'Confirm' button is positioned below the password field. At the bottom right of the page, there is a small link that says 'Elenco di lettura'.

Figure 4: Policy Maker Sign Up

The entered code determines the type of user trying to register, so depending on the code the user will be directed to a distinct sign-up page specific to his category with a different form to fill in.

The new screen shows a series of fields to be filled in, of which three, namely "First Name", "Last Name" and "Password" are mandatory for each user who wants to register and respectively indicate name, surname and the password to be set in order to access the system.

Therefore, the implementation choice made was to allow the registration to any suitable user even if only the farmers functionalities have been developed. Depending on the type of user, additional information may be requested as shown (Fig.2, Fig.3, Fig.4), all information must be entered in order to complete the registration correctly.

After completing the completion of the fields described above, if the user is a farmer is directed to a screen (Fig.5) aimed at registering their farm in the system. This new screen shows the user three new fields to fill in "Address", "Phone" and "N Lands", which respectively indicate the address of the farm to be registered, its telephone number and the number of lands it owns.

During the registration phase, the farmer is asked to indicate only the quantity of land owned because these are by default registered in the system as empty lands, it will then be up to the farmer after his first access, the task of correctly setting the features of each land owned and keep them updated over time.

At the end of the registration process the user is sent to a screen (Fig.6) that confirms the correct registration and communicates his ID (which we remember to be assigned by the system) which he will need, together with the password, to access the system via login.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ITD/CreateUser.do'. The browser's toolbar includes various icons for applications like Drive, Gmail, Maps, YouTube, LinkedIn, Yahoo, PolMail, Portale servizi, WeBeep, POLIMI chat, and Chess. The page features a dark red header with the 'DREAM' logo in white. Below the header, the main content area has a white background with the heading 'Enter your farm data'. Under this heading, there are three input fields labeled 'Address:', 'Phone:', and 'N° Lands:'. A green 'Confirm' button is positioned below the input fields.

DREAM

Enter your farm data

Address:

Phone:

N° Lands:

[Confirm](#)

Figure 5: Farm Sign Up

The screenshot shows the same web browser window as Figure 5. The page now displays a green success message box. The message reads: 'Registration completed! Welcome on board, from this moment you are present in the system and you will be able to use all the services after having logged in. Remember to write down this ID code that has been assigned to you by the system and that you will use to log in together with the password. Your ID is 6061934901'. Below the message box is a 'Log in' button.

DREAM

Registration completed!

Welcome on board, from this moment you are present in the system and you will be able to use all the services after having logged in.

Remember to write down this ID code that has been assigned to you by the system and that you will use to log in together with the password.

Your ID is **6061934901**

[Log in](#)

Figure 6: Success Sign Up

As for the log in, it is shown on the home screen and requires the user to enter the ID and password in order to access the system; if one or both of the entered values are incorrect, the user validation fails and therefore the following error message is shown "Wrong ID or password".

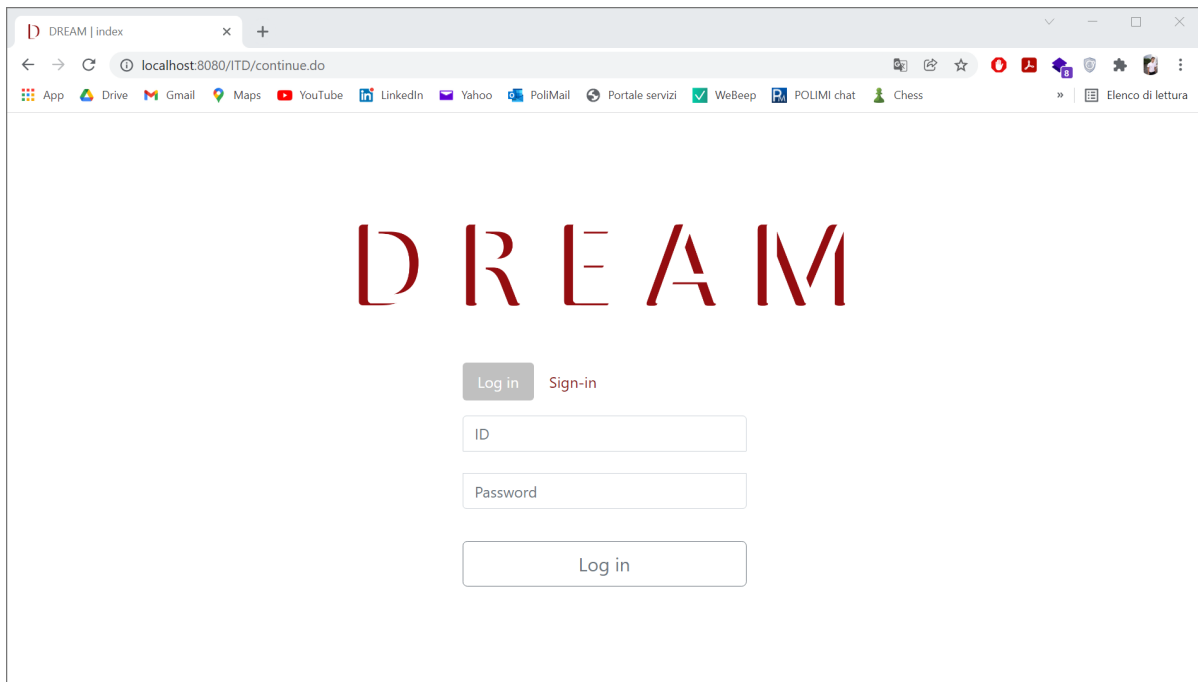
A screenshot of a web browser showing the login page for a system named "DREAM". The browser's address bar shows "localhost:8080/ITD/continue.do". The page features the word "DREAM" in large, red, serif capital letters. Below the title, there are two buttons: "Log in" (grey) and "Sign-in" (white with a red border). Under these buttons are two input fields labeled "ID" and "Password". At the bottom of the form is a "Log in" button. The browser's toolbar shows various icons and a search bar.

Figure 7: Log in

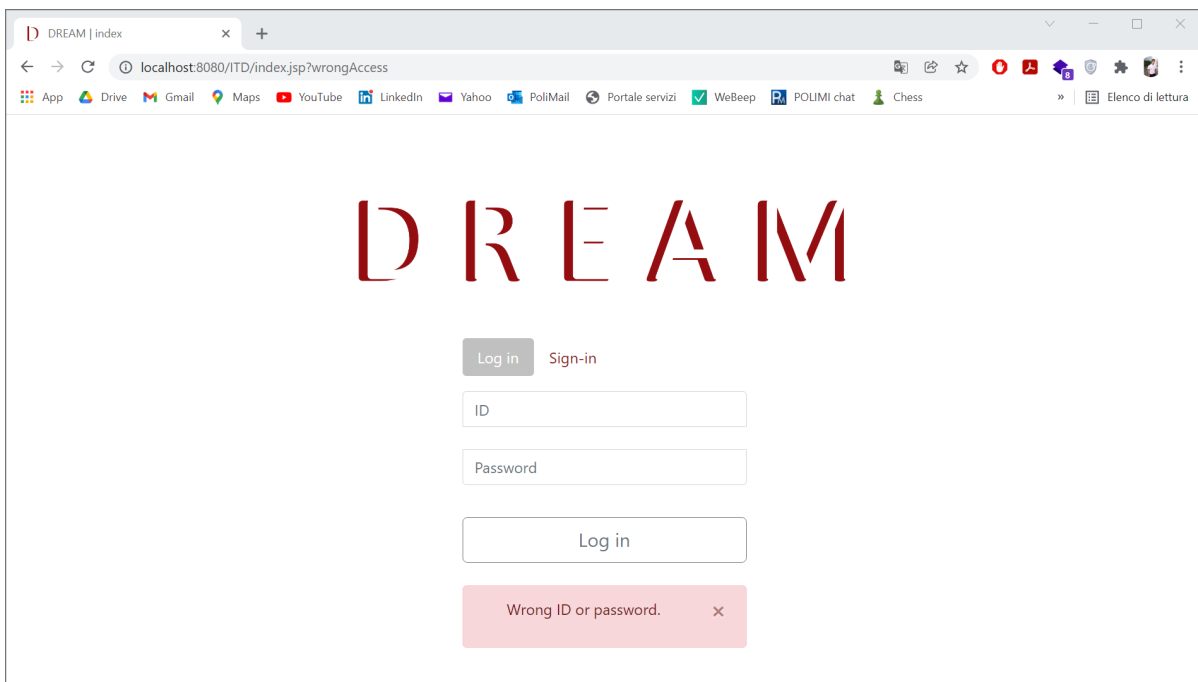
A screenshot of the same web browser showing the login page, but with an error message. The address bar now shows "localhost:8080/ITD/index.jsp?wrongAccess". The "DREAM" title and login form are identical to Figure 7. However, a red rectangular error message box is displayed at the bottom of the form, containing the text "Wrong ID or password." and a small "x" icon to close it.

Figure 8: Wrong Login

In the event that the user credentials requesting to access are correct, they are leads to their homepages.

2.2 Homepage

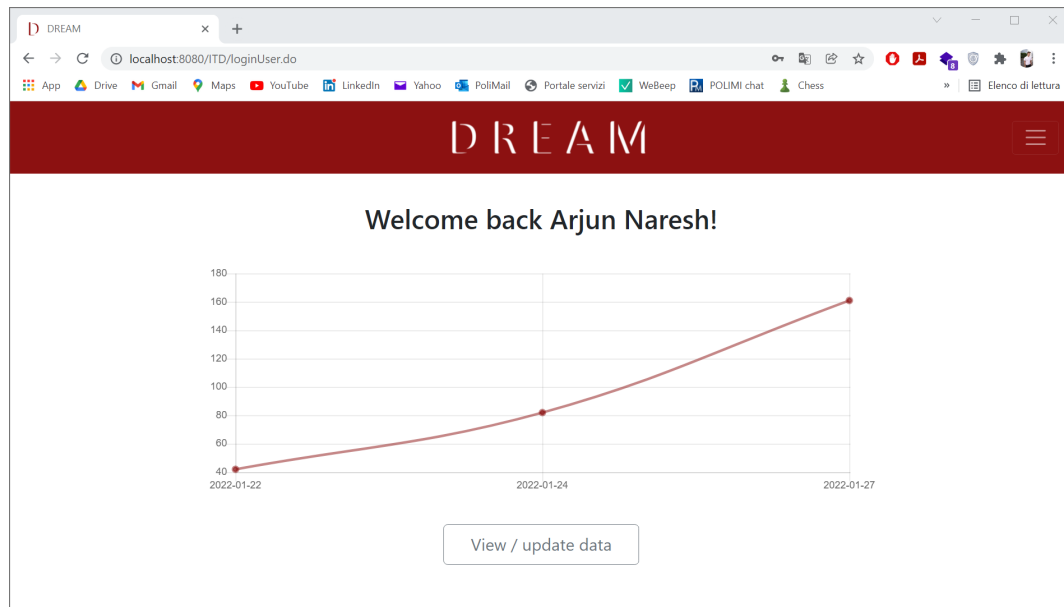


Figure 9: Farmer Homepage

Here the farmer's homepage is shown (Fig.9), where immediately under a welcome message there is a graph indicating the score associated with each crop that was carried out. Above the graph, at the top of the displayed screen, there is a navbar from which it is possible to access all the system functions, while below the graph there is the "View / update data" button connected to the data display and modification functions.

As for the homepages of agronomists and policy makers (Fig.10) a welcome message is shown, communicating to the user the fact that all the features related to his category have not been implemented and are assumed to work properly.

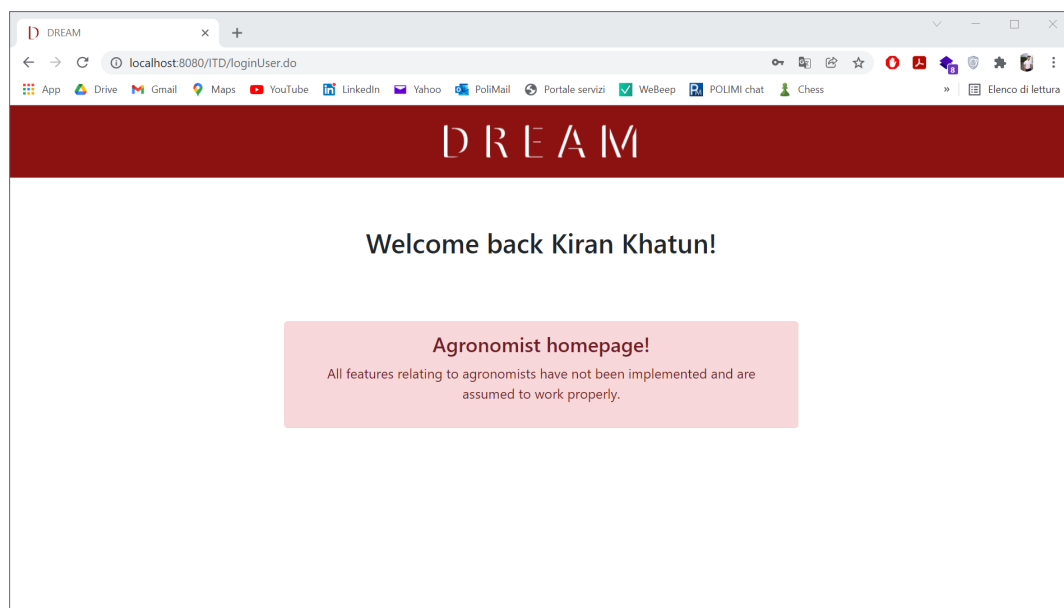


Figure 10: Agronomist Homepage

2.3 Navbar

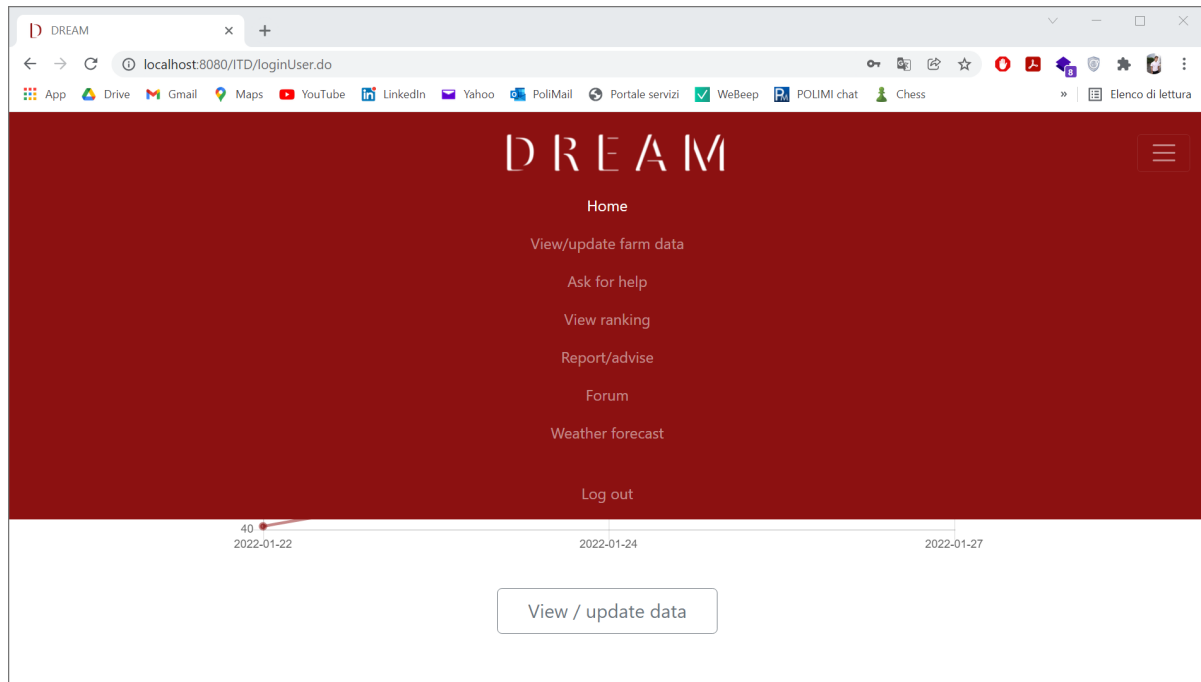


Figure 11: Navbar

The navbar consists of a burger button which when pressed activates a pop-up menu that allows the user to browse the website. Due to its function, this component is present in all the screens of the website.

From this screen we can therefore see all the main features that have been developed for the farmer.

2.4 View/update farm data

This feature is one of the most important and main of the system. It allows you to manage the collection of data and its visualization.

G5	Allow farmers to visualize relevant data based on their location and type of production
D1	Meteorological forecasts are always available
D2	The information inserted about the farm are reliable
D3	The information inserted about the farm are updated daily
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
R11	The system must show the ranking of the farms
R17	The system must display the data entered by the farmer in chronological order from the most recent to the least
R18	Farmers must be able to view their farms' production data
R19	Farmers must be able to view their score

G6	Allow farmers to keep track of their production data
D2	The information inserted about the farm are reliable
D3	The information inserted about the farm are updated daily
D4	The sensors deployed on the farmers territory and measuring the humidity of soil work correctly
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
D8	Farmers use the counter to measure the amount of water used and it works correctly
R16	The system must store the data entered by the farmer in chronological order from the most recent to the least
R20	Farmers can update their farms' production data
R21	The system must update the score associated to a farm every time its data are updated
R22	The system must update the ranking of the farms
R23	DREAM AI has to reprocess the projections and suggestions associated to each farm when its data are updated
R31	The system must send a notification to the farmers (at 21.00) to remind them to update the daily data

In this main screen (Fig.12) you can find all the main information about the farm. In particular, the first part contains the graph already present and explained in the home section, concerning the score of the crops carried out.

Second part concerns general information, such as the water used and the agronomist of reference.

Third and last part is dedicated to displaying the status of the fields present on the farm.

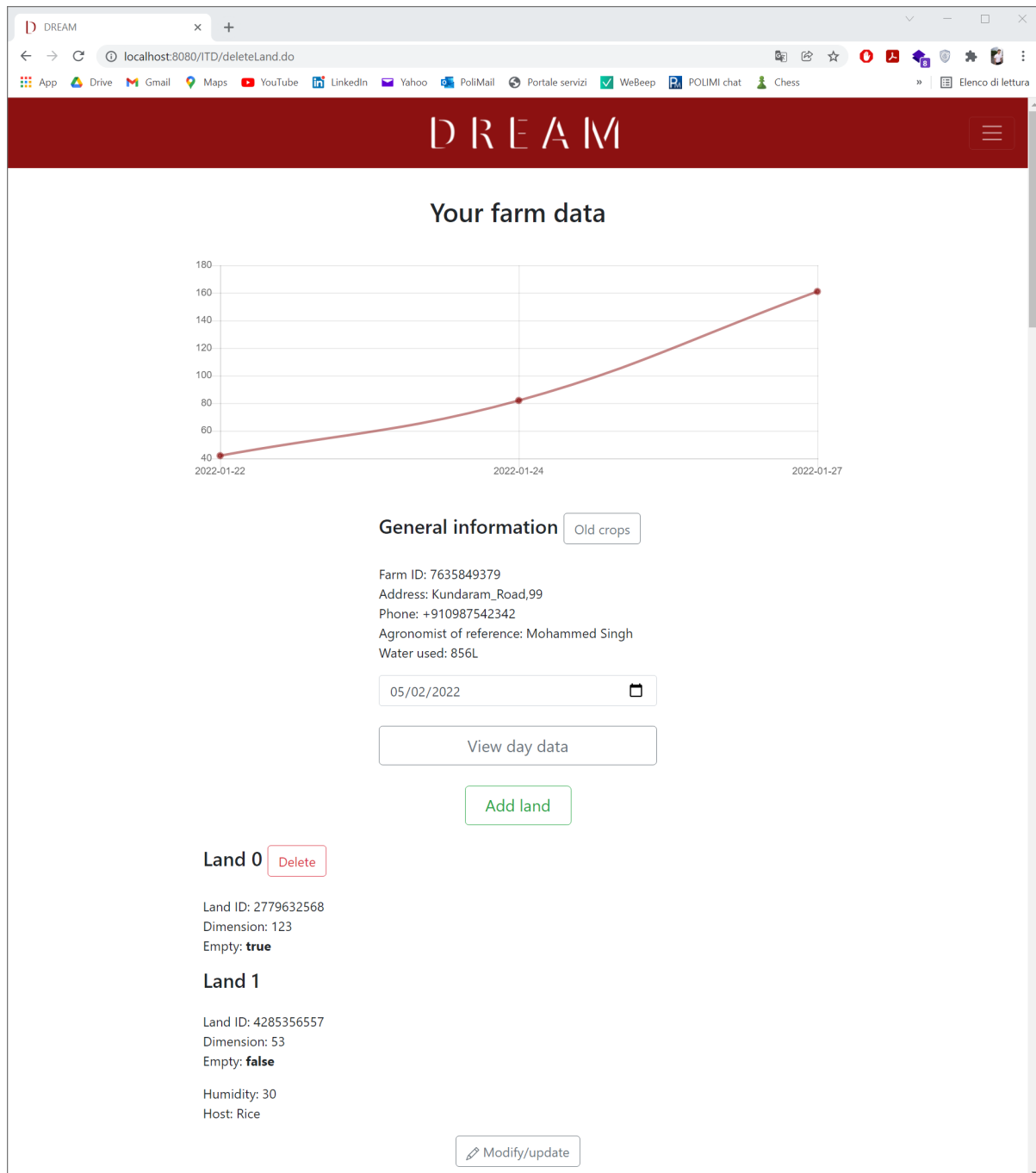


Figure 12: View Farm Data

The features on this page are many and will be explained one at a time.

2.4.1 Old crops

This feature allows you to view (Fig.13) all the crops made by the farm regardless of the land of reference, sorted in chronological order.

In particular, the information reported concerns the date, the product, the quantity and the crop score.

Date	Product	Quantity	Crop score
2022-01-27	Anise	8539	161
2022-01-24	Tea	10088	82
2022-01-22	Gourds	9058	42

Back

Farm ID: 7635849379
Address: Kundaram_Road,99
Phone: +910987542342
Agronomist of reference: Mohammed Singh
Water used: 856L

05/02/2022

View day data

Figure 13: Old Crops

2.4.2 View day data

This feature allows you to view the status and number of lands on a specific day. If you select a day before the farmer's enrollment, or after today's date, no land will be displayed.

While the state of the lands is tracked every day, general information about the farm is universal, and does not depend on the date.

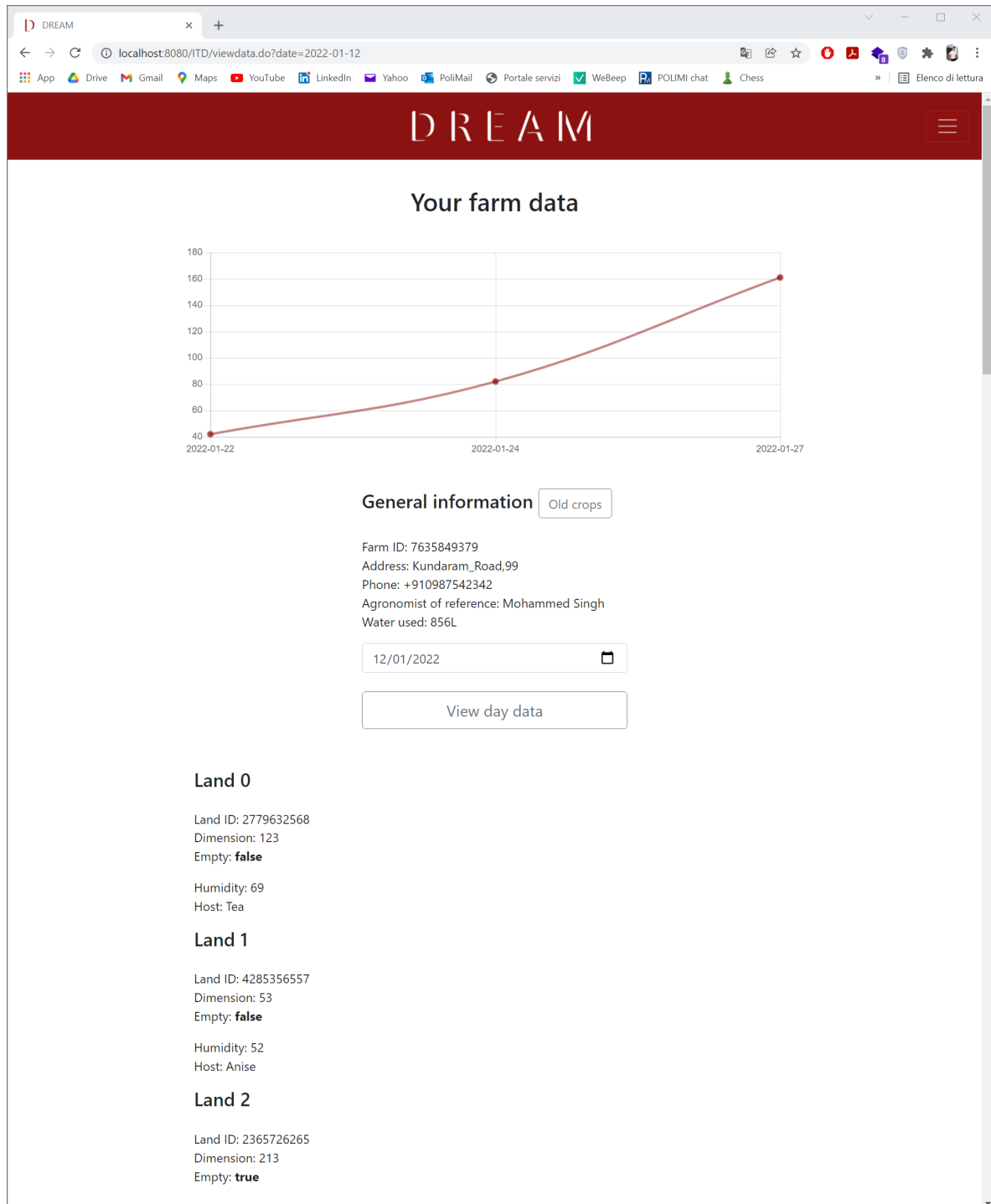


Figure 14: View Day Data

2.4.3 Modify/update data

With this feature you can update the data both relating to the general information of the farm and to the state of the land.

Important: the information can only be changed on the current day.

The screenshot shows a web browser window with the URL `localhost:8080/ITD/modifyupdate.do?Day=2022-02-05`. The page has a dark red header with the word "DREAM" in white. Below the header, the title "Modify/update farm data" is centered, followed by the date "2022-02-05".

The form is divided into three main sections:

- General information:** Contains input fields for "Farm ID" (pre-filled with 7635849379), "Address" (Kundaram_Road,99), "Phone" (+910987542342), and "Water" (856).
- Land 0:** Features a pink "Empty" button. Below it are input fields for "Land ID" (2779632568), "Host" (a dropdown menu), "Dimension" (123), and "Humidity" (0).
- Land 1:** Contains input fields for "Land ID" (4285356557), "Host" (Rice), "Dimension" (53), and "Humidity" (30).

At the bottom of the form, there is a "Crop:" label followed by a text input field containing "Quantity in gr.". At the very bottom, there are two buttons: "Confirm" (green) and "Cancel" (red).

Figure 15: Modify Update Data

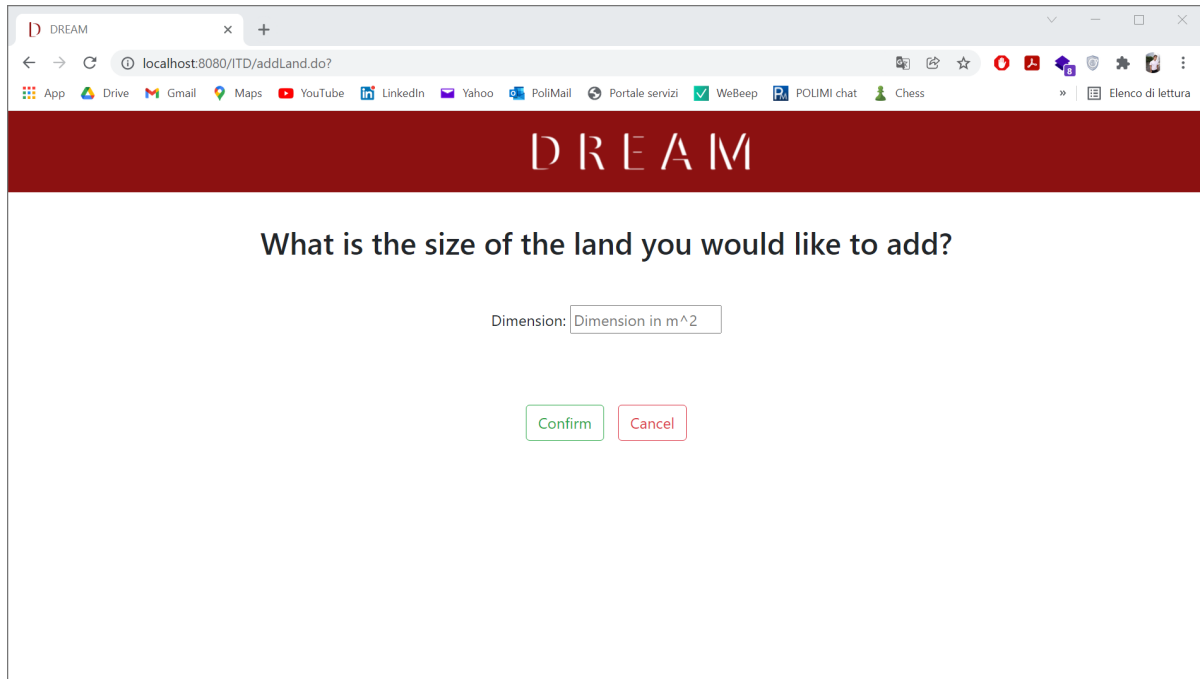
As you can see, if the land is empty you can fill it (cultivate it) by selecting one of the products that the land will host from those present in the system, furthermore when you want to fill in a land you must also specify the percentage of humidity in the field (a value between 1 and 100).

The size of a field can always be changed, whether it is empty or not.

When you are growing a field it would be a good idea to update its humidity daily, if you forget it, the data of the last humidity entered will be copied to the following day.

When you have collected the products of a land, just enter the quantity collected in grams, the information and the relative scores will automatically be updated and the land will automatically become empty.

2.4.4 Update/delete land



The screenshot shows a web browser window with the URL `localhost:8080/ITD/addLand.do?`. The page has a dark red header with the word "DREAM" in white. Below the header, the text "What is the size of the land you would like to add?" is displayed. Underneath, there is a label "Dimension:" followed by a text input field containing the placeholder text "Dimension in m^2". At the bottom of the form, there are two buttons: a green "Confirm" button and a red "Cancel" button.

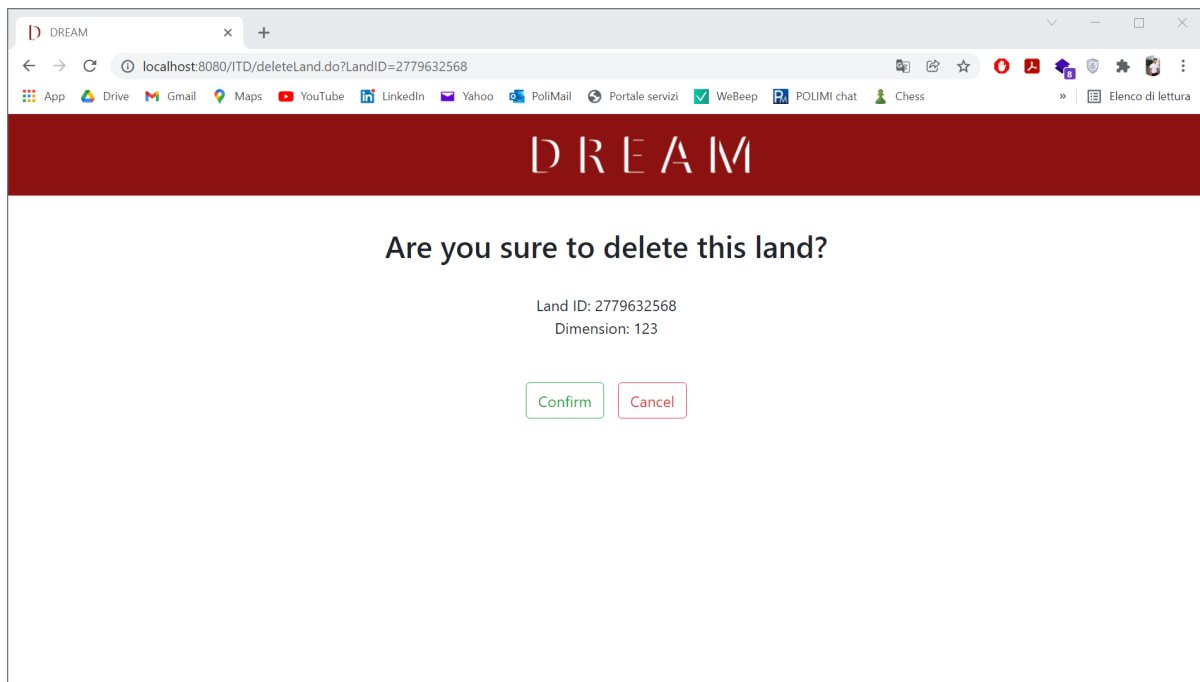
Figure 16: Add Land

This feature allows you to add or delete lands on your farm.

The changes are made on the current day and it is not possible to change past or future data.

Only empty lands can be eliminated.

At the request to insert a new land, the size will be requested and empty will be created by default.



The screenshot shows a web browser window with the URL `localhost:8080/ITD/deleteLand.do?LandID=2779632568`. The page has a dark red header with the word "DREAM" in white. Below the header, the text "Are you sure to delete this land?" is displayed. Underneath, the text "Land ID: 2779632568" and "Dimension: 123" is shown. At the bottom of the form, there are two buttons: a green "Confirm" button and a red "Cancel" button.

Figure 17: Delete Land

2.5 Ask for help

Another important feature implemented and made available to connected farmers is the ability to compile and save a request for help in the system to be forwarded then to competent agronomists. By implementing only the services related to farmers, it was decided to simplify the functionality with respect to what is described in RASD and DD; in fact, when the request is stored, the entire process is concluded and neither notification is sent.

G2	Allow farmers to ask the help they need
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
D9	The agronomist is available during working days
D10	The agronomist responds promptly to requests for help
R6	Agronomists must be notified if their help has been required
R7	The system must allow agronomists to view the data needed to contact farmers
R8	Agronomists must be able to access to sensitive data of other users

This service is accessed by clicking on the "Ask for Help" item in the navbar available on the home-page of each connected farmer. After clicking, the user is directed to a new screen dedicated to filling out the request for help; in this screen there are two fields to fill in: "Request Subject" and "Problems description".

The first, as easily understood, involves the insertion of the subject of the whole request, it is therefore a string of few characters with which to frame the reason and scope of the intervention requested from the recipient agronomist. The next field to be filled in allows farmers to better express their difficulties and problems through the possibility of inserting a greater number of characters in a specific textarea. Finally, the "Send Request" button is available which, once the request has been filled in, allows it to be uploaded in the system.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ITD/askforhelp.do?'. The browser's toolbar includes various icons for applications like Drive, Gmail, Maps, YouTube, LinkedIn, Yahoo, PolMail, Portale servizi, WeBeep, POLIMI chat, and Chess. The page header is a dark red bar with the word 'DREAM' in white capital letters and a hamburger menu icon on the right. The main content area has a white background with the title 'Ask for help' centered. Below the title, there are two input fields: 'Request Subject:' with a text input box containing the placeholder 'Enter title', and 'Problems description:' with a larger text area containing the placeholder 'Enter a description of problem you have'. At the bottom right of the form, there is a green 'Post' button.

Figure 18: Ask For Help

Once the request has been completed and loaded into the system, the farmer is redirected to his homepage and a special alert message is displayed to warn him about the actual saving of the request made, as well as the success of the entire process performed.

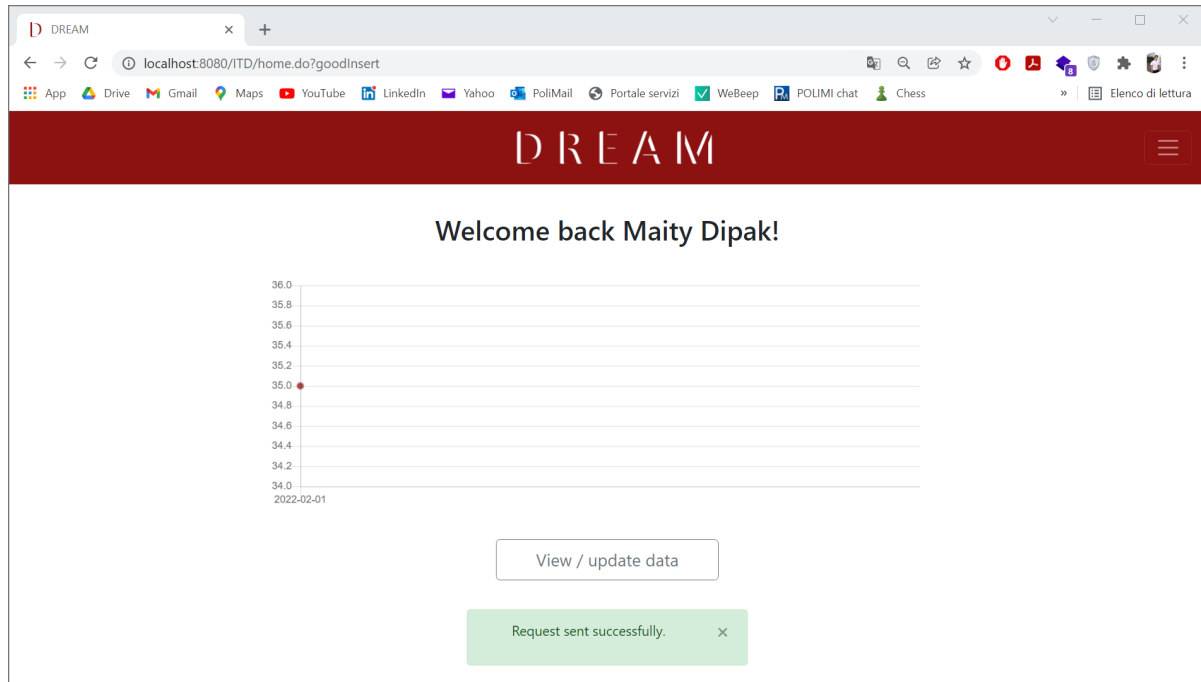


Figure 19: Correct send ask for help

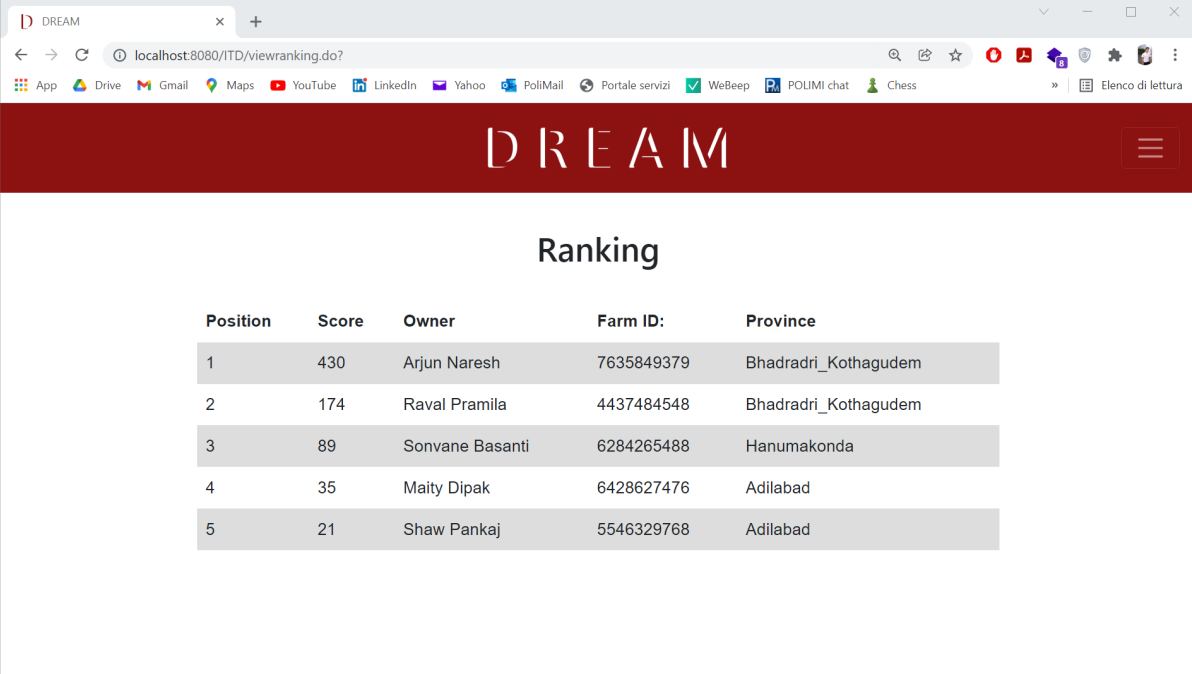
2.6 View ranking

Another important feature that has been implemented consists in the possibility offered to each farmer to consult the status of the ranking of the registered farms at any time, thus being able to view the position of his farm and the associated score.

The process of updating of the ranking is instead a process operated by the system and the user is kept in the dark. For the sake of simplicity, unlike what is described in RASD and DD, the update is not performed automatically once a minute but only when necessary, i.e. when new farms are added to the system or when the score of some already registered farms is updated.

G5	Allow farmers to visualize relevant data based on their location and type of production
D1	Meteorological forecasts are always available
D2	The information inserted about the farm are reliable
D3	The information inserted about the farm are updated daily
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
R11	The system must show the ranking of the farms
R17	The system must display the data entered by the farmer in chronological order from the most recent to the least
R18	Farmers must be able to view their farms' production data
R19	Farmers must be able to view their score

This service is accessed by clicking on the "View ranking" item in the navbar available on the home-page of each connected farmer. After clicking, the user is directed to a new screen where the ranking is displayed; specifically, this is a screen where for each position of the ranking the id of the farm in that position, the name and surname of its owner, its province of belonging and its total score are shown.



Position	Score	Owner	Farm ID:	Province
1	430	Arjun Naresh	7635849379	Bhadradi_Kothagudem
2	174	Raval Pramila	4437484548	Bhadradi_Kothagudem
3	89	Sonvane Basanti	6284265488	Hanumakonda
4	35	Maity Dipak	6428627476	Adilabad
5	21	Shaw Pankaj	5546329768	Adilabad

Figure 20: View Ranking

2.7 Report/advise

This is the service thanks to which connected farmers have the ability to view all the various reports recorded in the system written by the agronomists responsible for their farm. For simplicity and deciding to model only the functions of the system affecting farmers and not those relating to agronomists, it was decided to develop only the visualization of the reports without dealing with their drafting and their insertion into the system, aspects that we have instead given for discounted.

G5	Allow farmers to visualize relevant data based on their location and type of production
D1	Meteorological forecasts are always available
D2	The information inserted about the farm are reliable
D3	The information inserted about the farm are updated daily
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
R11	The system must show the ranking of the farms
R17	The system must display the data entered by the farmer in chronological order from the most recent to the least
R18	Farmers must be able to view their farms' production data
R19	Farmers must be able to view their score

This service is accessed by clicking on the "Report / advise" item in the navbar available on the

homepage of each connected farmer. After clicking, the user is directed to a new screen where he is offered two alternative possibilities, the possibility of viewing all the reports recorded in the system for his farm or only those relating to a specific day.

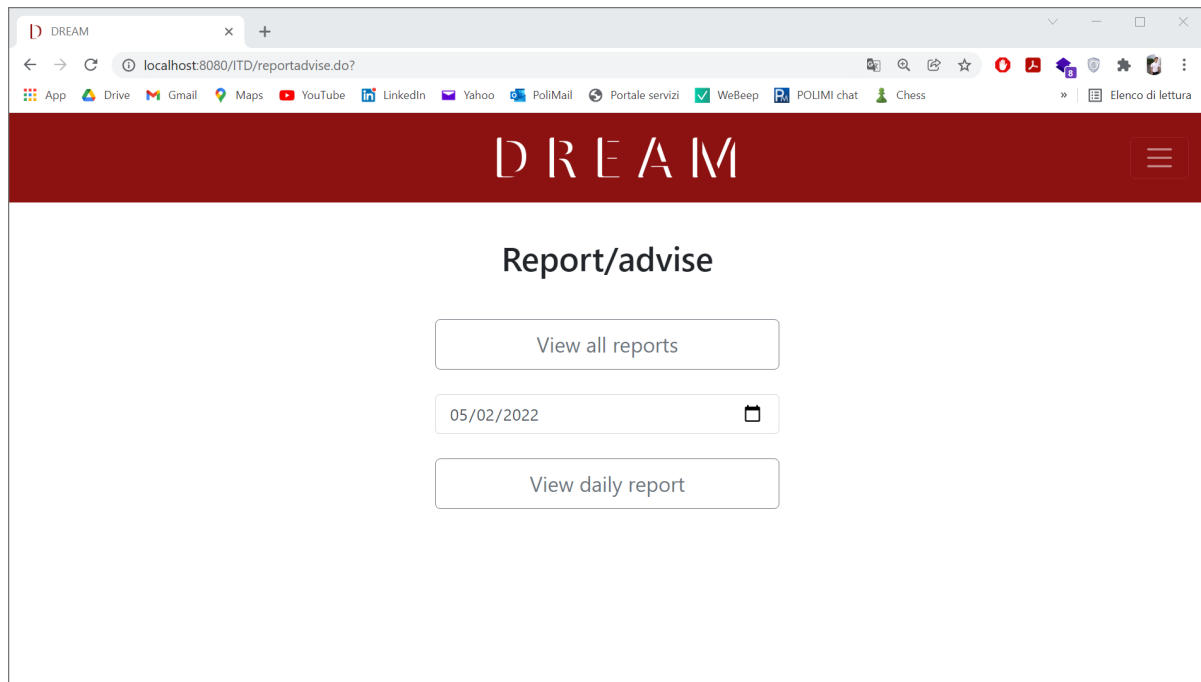


Figure 21: Report Selection

The first option can be reached by pressing the appropriate "View all reports" button, through which the user is directed to a new screen where the list of all the reports made on his farm is shown (Fig.22); specifically, the reports are displayed from the most recent at the top of the list to the oldest at the bottom and for each report its identifier, the ID of the farm and the agronomist involved, its date, the advice provided by the agronomist and a brief description of the problems encountered are shown.

Instead of viewing all the reports on his farm, the connected user can choose to filter them by date by selecting a certain date and pressing the "View daily report" button, in so doing he will be directed to a new screen where only the reports loaded at the date indicated in the system will be shown. In this screen the reports of the chosen day are represented in the same way as the screen reached by selecting "View all reports".

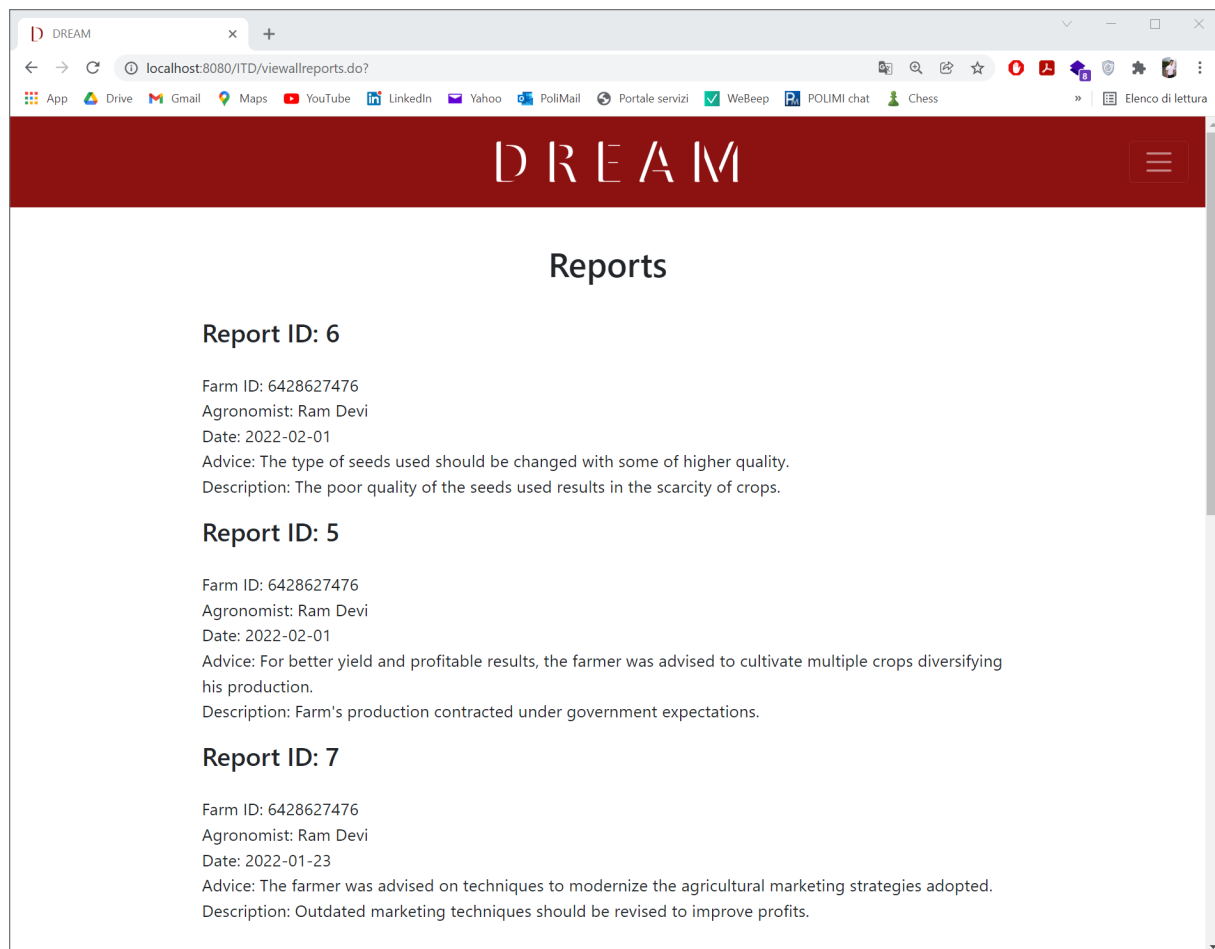


Figure 22: View All Reports

2.8 Forum

This feature allows farmers to interact via discussions that can be commented on.

G7	Allow farmers to create discussion forums with the other farmers
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
R24	The system must allow farmers to create a new discussion on the forum
R25	A farmer must be able to comment on a forum discussion
R26	A comment from a user must be able to tag another user
R27	The system must send a notification to a user when when his discussion is commented on
R28	The system must send a notification to a user when tagged in a comment
R29	The forum must store the discussions/comments in chronological order
R30	The forum must display the discussions/comments in chronological order

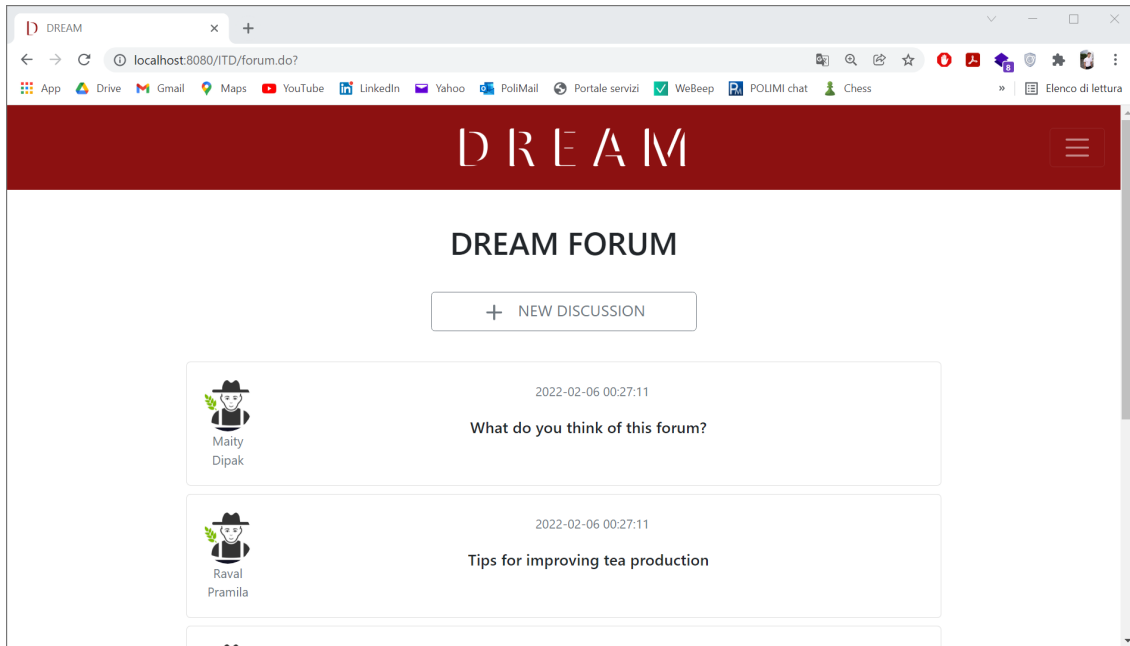


Figure 23: Forum

The first screen (Fig.23) shows all the discussions already present in the system, but only the title, the farmer who created it and the creation date / time of them. From here the user can scroll to find a discussion topic that interests him, or press the "New discussion" button to create a new discussion.

If this last action is performed, a screen will appear (Fig.24) asking the user to enter the subject and body of the new discussion before confirming. If either of these two fields is empty, the entry will not take place.

If, on the other hand, the user selects a discussion, a screen will appear (Fig.25) with the details on the discussion (the body) and the related comments. Also there is a section that allows the user to comment on the specific discussion.

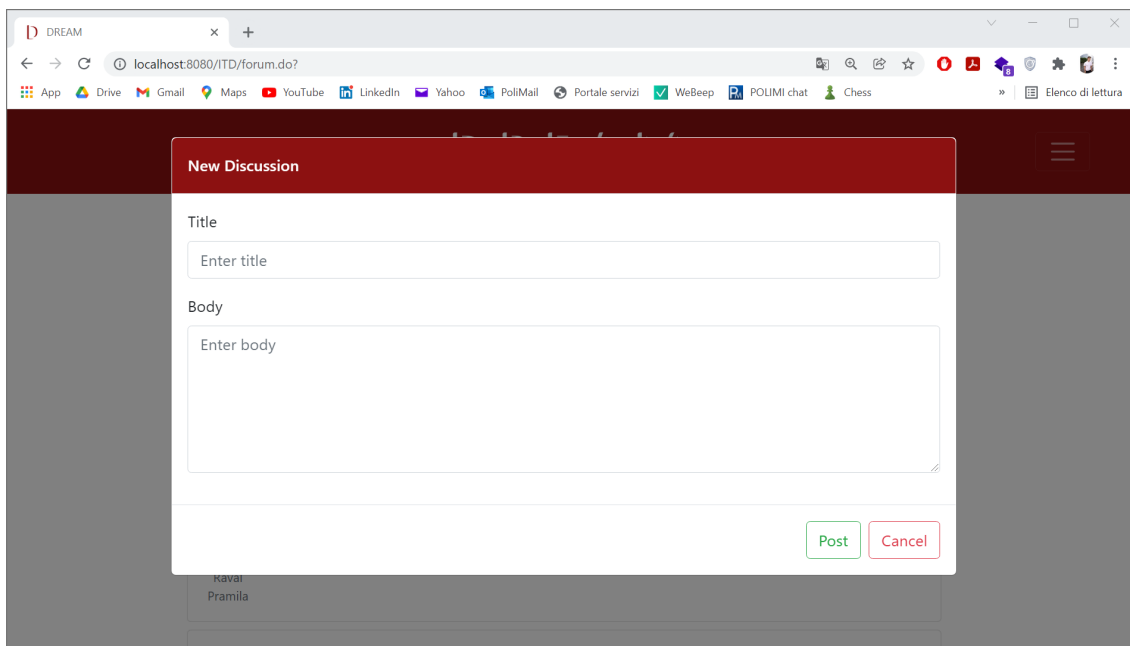


Figure 24: New Discussion

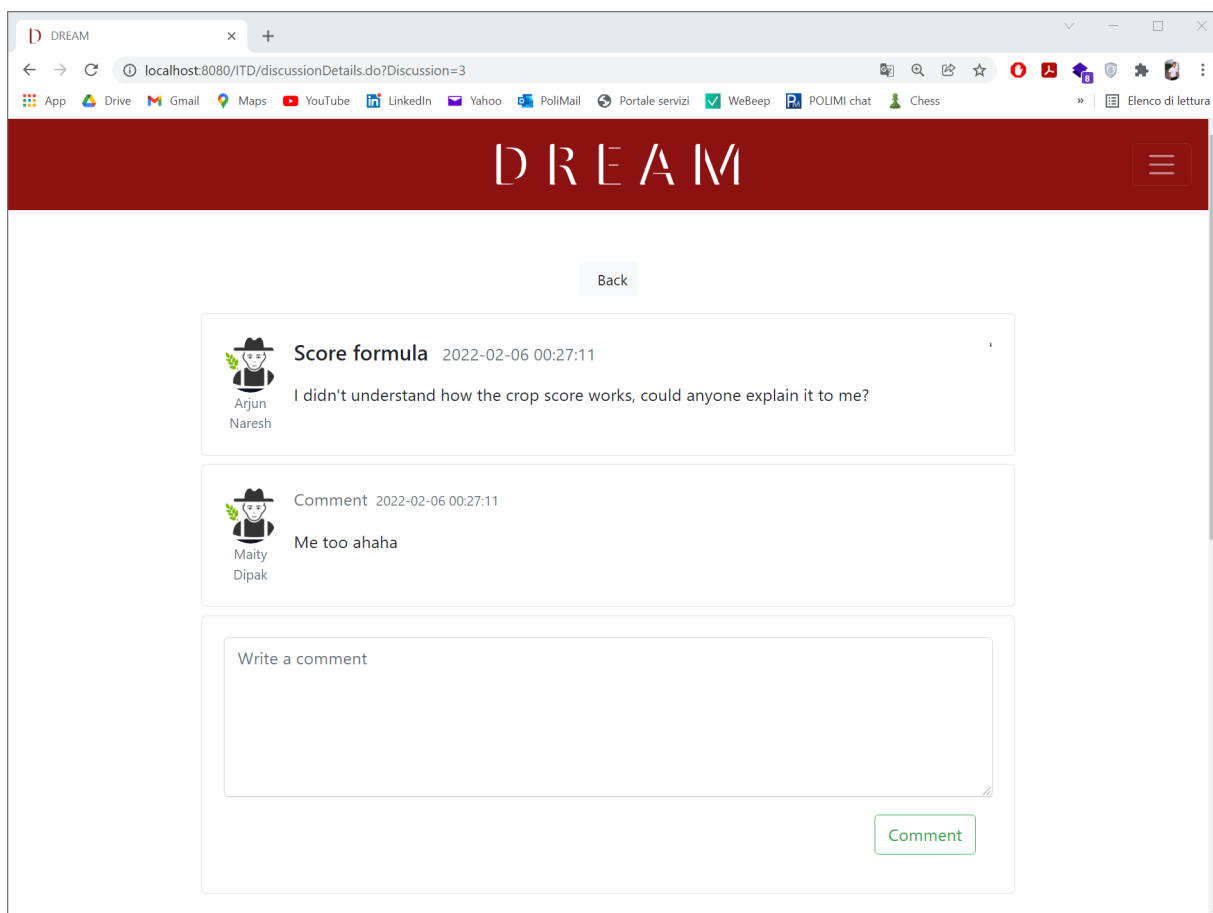


Figure 25: Discussion Detail

2.9 Weather forecast

This DREAM service allows connected farmers to view weather forecasts at any time by simply redirecting them to the TSDPS site. This is a feature which contributes to the achievement of the following goal and to the implementation of the following requirements:

G4	Being able to anticipate climatic phenomena
D1	Meteorological forecasts are always available
D6	Each farm has internet connection and a device to access the platform (smart-phone or pc)
D7	Each user has the necessary skills to navigate the designed user interface and use its features
R14	The system must allow users to be redirected to TSDPS meteo

This service is accessed by clicking on the "Weather forecast" item in the navbar available on the homepage of each connected farmer. After clicking, the user is redirected to the site of the TSDPS where he has the possibility to consult the weather forecast and all the desired information.

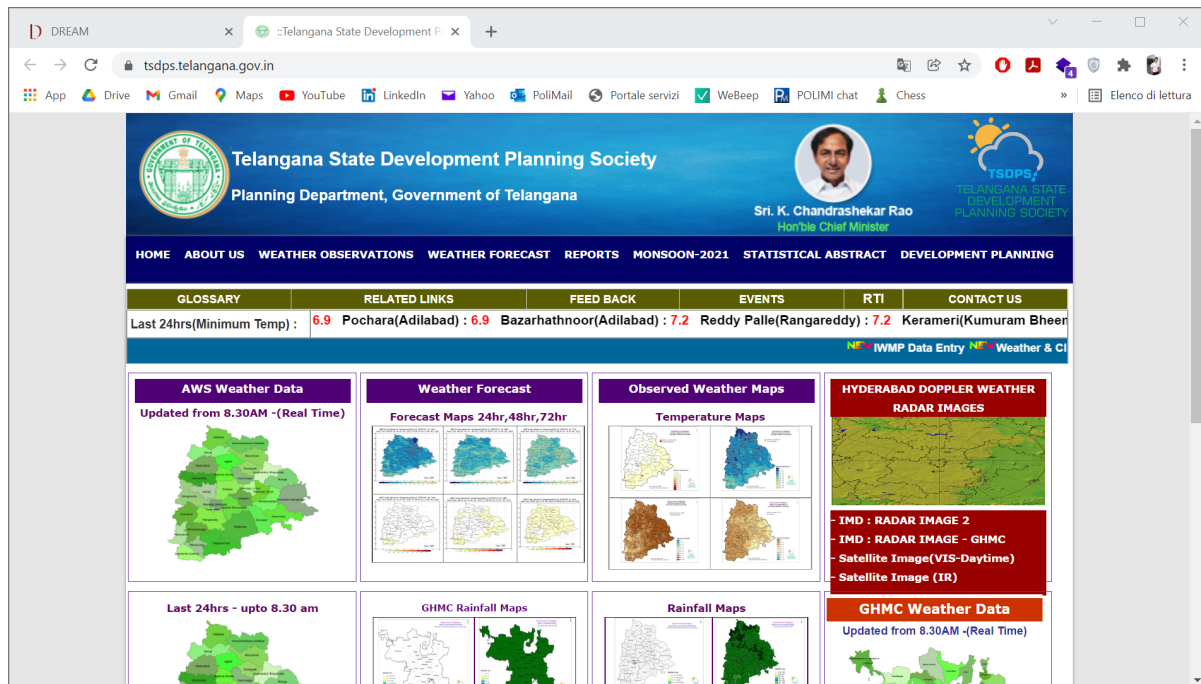


Figure 26: TSDPS

2.10 Log out

This function is accessed by clicking on the "Log out" item in the navbar available on the homepage of each connected farmer and works as easily understood in the opposite way to the login process described above. After clicking, the user is sent back to the initial page of the system, his session is expired and he will have to log in again to keep using the functionalities of the system.

3 Prototype limitations

In this section we want to expose some of the main limitations present in our prototype, which are not expected to be present also in the final version of the project.

3.1 Security

The issue of security is very important in the implementation of the project, as this is used for government use. We have tried to give a general idea of the importance of this aspect in our prototype, allowing the registration only to people in possession of a registration code, assigning from the system a unique 10-digit code which is used to log in to the user, or even the implementation of a session that expires after one hour of access.

Despite these functions, others have been left out, such as encrypting data in the database and server-side validation of data entered by a user (most of the checks are done on the client side in the prototype, which is insecure). Some malicious users could easily bypass some security checks and break into the system, which is why it is very important to increase the security levels in the final development.

3.2 Score

The calculation of the score has been enormously simplified in the realization of this prototype. In fact it is calculated by carrying out a simple relationship between the quantity collected and the size of the field. This formula in a real system would lead to an advantage in certain farms over others, so it is very important that in a final development the score is calculated as described in the RASD, also taking into account the harvested product and the farm area.

3.3 Forum

There has also been a simplification with regard to the forum. This feature was added to give an idea but logically in a future development it foresees that in addition to the insertion of discussions and comments also their possible cancellation by the creators.

3.4 Ranking

As for the ranking update, this happens when the score is changed in our prototype. This mechanism works correctly as long as there are few users who connect simultaneously to the server and therefore also the operations to modify the score are consequently. However, imagining a more large-scale application of the project these modification operations could slow down the server excessively and therefore it would be better to update the ranking once a minute, as already conceived in the DD.

3.5 Land daily update

In the final version we have already described how there should be a function that updates the land data at midnight, copying the information for the next day, before these are further modified / updated.

In the prototype this function does not exist as it is not designed to remain active on a server continuously for days. For this reason the SetupDatabase will simulate the operation of the database until the day the prototype is opened, if the server is accessed the next day there will be inconsistencies, so it is recommended to delete the database and let it be rebuilt.

4 Development frameworks adopted

The prototype of DREAM created as already mentioned was developed as an Eclipse JEE Dynamic Web Project; therefore, the framework used for the development of the project was JEE and most of the project files were written in Java separated between beans, servlets and entities.

The project was run on a Tomcat server installed and configured in Eclipse.

It was decided to use a relational database and as a DBMS we chose to use PostgreSQL, a free and open-source relational database management system (RDBMS) that emphasizes extensibility and SQL compliance; to manage the DBMS we use pgAdmin, the most popular and feature rich Open Source administration and development platform for PostgreSQL. In addition, the JPA framework was used, through which it was possible to map the relational database tables into java classes called entities and their columns into attributes of these classes. To perform queries on the database, where possible, the functions offered by JPA have always been used; while in a few other cases, for simplicity, JDBC, namely a Java-based data access technology used for Java database connectivity, have been used. JPA was used through the popular Hibernate framework, an object – relational mapping tool for the Java programming language, whose configuration file can be found in the project.

It was decided to create for each functionality offered by the system to users, a bean equipped with functional methods for satisfying this service and a specific servlet that uses the bean's methods to provide answers to the requests received from users.

Beans contain the business logic of the entire application, through the methods they are equipped with, they manipulate entities to perform operations. The servlets, on the other hand, receive user requests in the form of HTTP requests from JSP pages, process the appropriate responses and provide the responses processed as HTTP requests to the other JSP pages. The user interface is represented by these JSP pages containing elements of HTML, JavaScript and CSS; furthermore, in their development Bootstrap was used, namely a free and open-source CSS framework directed at responsive, mobile-first front-end web development. Further details about the frameworks used and about what is necessary in general to install and run the project correctly are shown in chapter 6, while the structure of the source code will be explained and analyzed in more detail in the next chapter.

As for the tests, the JUnit5 unit testing framework was used together with Mockito. Mockito is an open source testing framework for Java released under the MIT License. The framework allows the creation of test double objects in automated unit tests for the purpose of test-driven development or behavior-driven development.

5 Source code structure

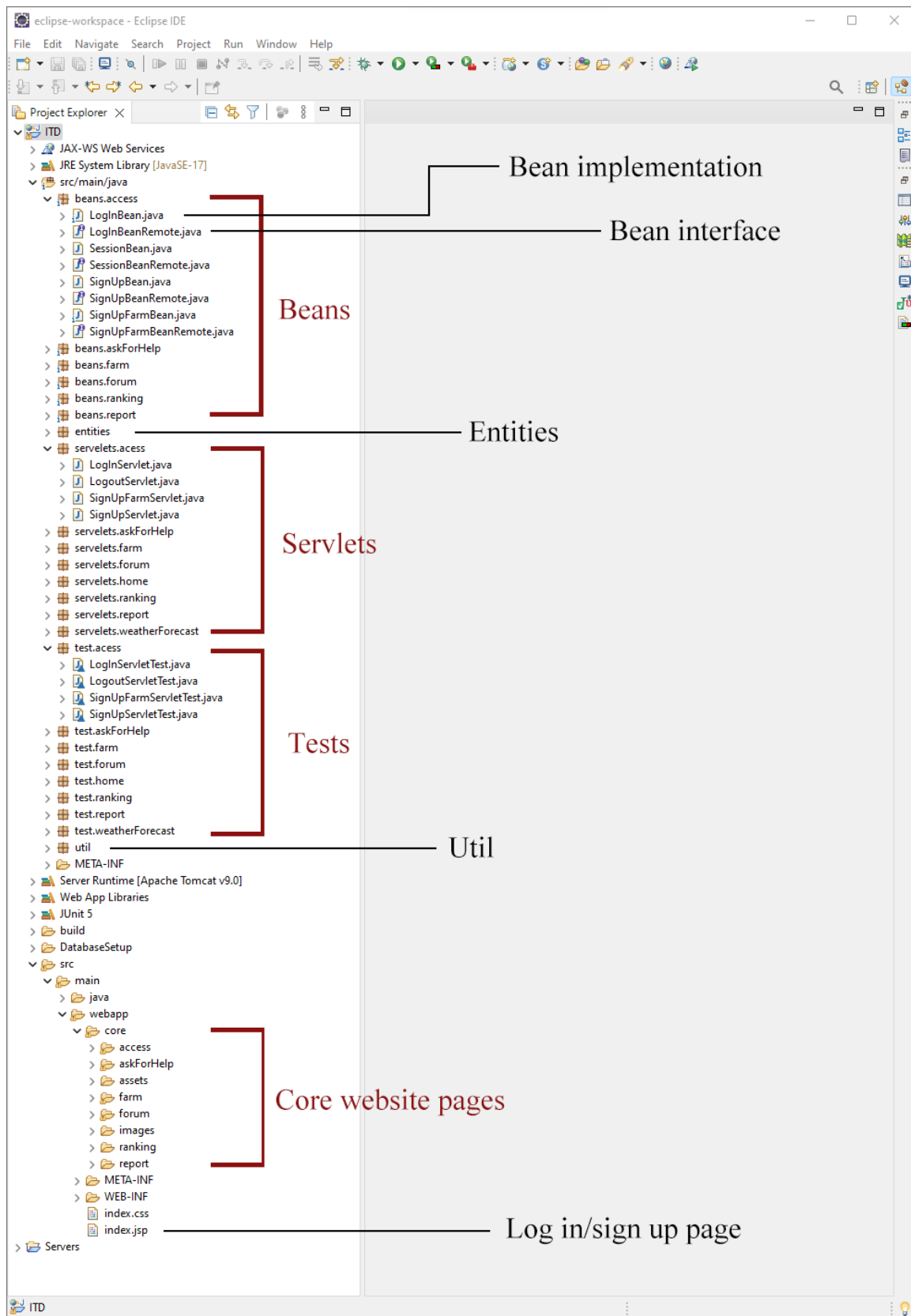


Figure 27: Source Code Structure

In general, each section has been divided further on the functionality it refers to. The features are those already listed above, namely:

- **Access:** log in, sign up and session management.
- **Ask for help:** ask for help management.
- **Farm:** management of the visualization and updating of farm data.
- **Forum:** forum management.
- **Ranking:** ranking management.
- **Report:** management of report view.

For each bean, both an interface and an implementation class have been created.

The "entities" section collects all the database entities, so that they can be used with JPA.

"Util" is a section that contains some auxiliary classes such as SetupDatabase which has the task of creating the database when it does not exist and of populating it with initial data.

6 Test cases

This section will show all the main tests performed on the system. As already mentioned, the frameworks of JUnit5 and Mockito were used for the realization.

The basic idea was to simulate requests (instantiating them as mock objects) and verify the behavior of the servlets, this by analyzing one servlet at a time, then with unit tests. For this reason, test suites have been created for each servlet. For this reason, a test suite has been created for each servlet.

Servlets use beans to interface with the database, so some tests verify that the data in the database has been updated correctly, but in some cases the interaction with the database is not possible to test if it is correct using Mockito (for example the visualization), this is because it is possible to monitor only the interactions (method invocations) that occur from the request or response, since these are instantiated as mock objects do not behave like real objects and have limitations.

In addition to this type of test, numerous further informal tests were carried out by simulating a user's navigation through the web and monitoring the behavior of requests and responses with a proxy.

6.1 Access tests

6.1.1 LogInServlet test suite

testPost1

Test summary	Log in not performed with wrong credentials
Previous condition	
Post condition	Redirect to index.jsp page with error message alert
Execution record	True

testPost2

Test summary	Log in not performed with empty credentials
Previous condition	
Post condition	Redirect to index.jsp page with error message alert
Execution record	True

testPost3

Test summary	Login successful with correct credentials (farmer)
Previous condition	
Post condition	Redirect to farmer homepage
Execution record	True

testPost4

Test summary	Login successful with correct credentials (agronomist)
Previous condition	
Post condition	Redirect to agronomist homepage
Execution record	True

testPost5

Test summary	Login successful with correct credentials (policy maker)
Previous condition	
Post condition	Redirect to policy maker homepage
Execution record	True

6.1.2 LogoutServlet test suite

testGet1

Test summary	Log out request successful
Previous condition	User is logged into the system
Post condition	Redirection on the index.jsp page and the session associated with the user no longer present in the database
Execution record	True

6.1.3 SignUptServlet test suite

testPost1

Test summary	Wrong code does not allow to go to registration
Previous condition	
Post condition	Redirect to index.jsp page with error message alert
Execution record	True

testPost2

Test summary	Correct code takes you to the registration screen (farmer)
Previous condition	
Post condition	Redirect to signUpFarmer.jsp
Execution record	True

testPost3

Test summary	Correct code takes you to the registration screen (agronomist)
Previous condition	
Post condition	Redirect to signUpAgronomist.jsp
Execution record	True

testPost4

Test summary	Correct code takes you to the registration screen (policy maker)
Previous condition	
Post condition	Redirect to signUpPolicyMaker.jsp
Execution record	True

testPost5

Test summary	Registration code cannot be reused twice
Previous condition	
Post condition	Redirect to index.jsp page with error message alert
Execution record	True

testPost6

Test summary	Missing parameters in the registration phase do not allow you to proceed (farmer)
Previous condition	The user is on the registration screen
Post condition	The user remains on the registration screen
Execution record	True

testPost7

Test summary	Correct user registration (farmer)
Previous condition	The user's registration code is valid and the user is on the registration screen
Post condition	Registration was successful and the farmer is redirected to the farm registration page
Execution record	True

testPost8

Test summary	Correct user registration (agronomist)
Previous condition	The user's registration code is valid and the user is on the registration screen
Post condition	Registration was successful and the user is redirected to signUp-Completed.jsp
Execution record	True

testPost9

Test summary	Correct user registration (policy maker)
Previous condition	The user's registration code is valid and the user is on the registration screen
Post condition	Registration was successful and the user is redirected to signUp-Completed.jsp
Execution record	True

6.1.4 SignUpFarmServlet test suite

testPost1

Test summary	If fields are missing from the farm registration, the user remains on the registration page and the farm is not entered in the database
Previous condition	The farmer has already been registered
Post condition	The farm is not in the database and the user remains to signUpFarm.jsp
Execution record	True

testPost2

Test summary	The farm is successfully registered
Previous condition	The farmer has already been registered
Post condition	The farm is in the database and user redirected to signUpCompleted.jsp
Execution record	True

6.2 Home tests

6.2.1 HomeServlet test suite

testGet1

Test summary	The request to view the home page is successful
Previous condition	User logged in
Post condition	Redirect to farmerHomepage.jsp
Execution record	True

6.3 Farm tests

6.3.1 ViewDataServlet test suite

testGet1

Test summary	The request to view the home page is successful
Previous condition	User logged in
Post condition	Redirect to view_update.jsp
Execution record	True

6.3.2 UpdateDataServlet test suite

testGet1

Test summary	The request to view the home page is successful
Previous condition	User logged in
Post condition	Redirect to view_update.jsp
Execution record	True

testPost1

Test summary	Correct update of farm data
Previous condition	User logged in and is in the farm data update page
Post condition	The farm data has been successfully edited in the database and the user is redirected to his farm data view page
Execution record	True

6.3.3 AddLandServlet test suite

testPost1

Test summary	Correct insertion of a land
Previous condition	User logged and in view farm data screen
Post condition	Land present in the database and redirect to view farm data screen
Execution record	True

6.3.4 DeleteLandServlet test suite

testPost1

Test summary	Correct deletion of a land
Previous condition	User logged and in view farm data screen, at least one land in the user's farm
Post condition	Land no more present in the database and redirect to view farm data screen
Execution record	True

6.4 Ask for help tests

6.4.1 AskForHelpServlet test suite

testGet1

Test summary	The request to view ask for help page is successful
Previous condition	User logged in
Post condition	Redirect to ask_for_help.jsp
Execution record	True

testPost1

Test summary	The help request is not sent if not all fields are filled in
Previous condition	User logged in and is on the ask for help screen
Post condition	Data not entered in the database and the user remains on the ask for help page
Execution record	True

testPost2

Test summary	Correct insertion of the ask for help request
Previous condition	User logged in and is on the ask for help screen
Post condition	Request entered in the database and user redirected to the home page with success alert message
Execution record	True

6.5 Ranking tests

6.5.1 ViewRankingServlet test suite

testGet1

Test summary	The request to view the ranking page is successful
Previous condition	User logged in
Post condition	user redirected to the page view_ranking.jsp
Execution record	True

6.6 Report tests

6.6.1 ReportAdviseServlet test suite

testGet1

Test summary	View initial report screen
Previous condition	User logged
Post condition	Redirect to select_reports.jsp
Execution record	True

testGet2

Test summary	View all reports screen
Previous condition	User logged and in the initial report screen
Post condition	Redirect to view_all_reports.jsp with all reports
Execution record	True

testGet3

Test summary	View daily reports screen
Previous condition	User logged and in the initial report screen
Post condition	Redirect to view_all_reports.jsp with all daily reports
Execution record	True

6.7 Forum tests

6.7.1 ForumServlet test suite

testGet1

Test summary	The request to view the forum page is successful
Previous condition	User logged in
Post condition	Redirect to forum.jsp
Execution record	True

testPost1

Test summary	Inserting a discussion without body
Previous condition	User logged and in forum general page
Post condition	Discussion not inserted and redirect to forum.jsp
Execution record	True

testPost2

Test summary	From a post request to get request
Previous condition	User logged
Post condition	Redirect to forum.jsp
Execution record	True

testPost3

Test summary	Correct posting of a discussion
Previous condition	User logged in and in forum general page
Post condition	Discussion present in the database and redirect to forum.jsp
Execution record	True

6.7.2 DiscussionDetailsServlet test suite

testGet1

Test summary	Correct view of a discussion in detail
Previous condition	User logged in and in forum general page
Post condition	Redirect to discussion_details.jsp
Execution record	True

testPost1

Test summary	Insert an empty comment
Previous condition	User logged in and in forum general page
Post condition	Comment not insert in database and user redirect to forum general page
Execution record	True

testPost2

Test summary	Comment insertion successfully
Previous condition	User logged in and in forum general page
Post condition	New comment present in the database and redirected to forum general page
Execution record	True

6.8 Weather forecast tests

6.8.1 WeatherForecastServlet test suite

testGet1

Test summary	Correct redirection to the weather forecast site
Previous condition	User logged
Post condition	User redirect to https://tsdps.telangana.gov.in/
Execution record	True

7 Installation instructions

Versions of the tools used:

- Eclipse: 4.22.0 (<https://www.eclipse.org/downloads/>)
- Tomcat: 9.0.58 (<https://tomcat.apache.org/download-90.cgi>)
- PostgreSQL: 14.1 (<https://www.postgresql.org/download/>)
- pgAdmin: 6.1 (<https://www.pgadmin.org/download/>)

We decided to shoot a short video in which the installation instructions are explained step by step:
https://youtu.be/FKGL5_Xry34

Folder with all material:

<https://github.com/pietrovalente/DREAM-software-engineering-2/tree/main/DeliveryFolder/ITD>

For any problems, do not hesitate to contact us: pietro1.valente@mail.polimi.it , andrea.seghetto@mail.polimi.it

8 Effort Spent

Document writing

Topic	Pietro Valente Hours	Andrea Seghetto Hours	Total Hours
General discussion	together	together	7h
Introduction	1h	2h	3h
Functions implemented	7h	5.5h	12.5h
Prototype limitations	1h	0h	1h
Development frameworks adopted	1h	2.5h	3.5h
Source code structure	1.5h	1h	2.5h
Test cases	3.5h	5h	8.5h
Installation instructions	1.5h	1.5h	3h
Graphical layout	2h	0h	2h
Final revision of the document	4h	5h	9h
Effort tracking	together	together	2h
	31.5h	31.5h	54h

Implementation and testing

Topic	Pietro Valente Hours	Andrea Seghetto Hours	Total Hours
Sign up	3h	21.5h	24.5h
Log in	8.5h	12h	20.5h
Homepages	11h	2.5h	13.5h
Navbar	4.5h	0h	4.5h
View/update farm data	20.5h	0h	20.5h
Ask for help	0h	10.5h	10.5h
View/update ranking	2h	15h	17h
View reports	2h	11.5h	13.5h
Forum	21h	0h	21h

Topic	Pietro Valente Hours	Andrea Seghetto Hours	Total Hours
Weather forecast	0.5h	0h	0.5h
Log out	0.5h	0.5h	1h
Mockito functioning	5h	5h	10h
Access tests	7h	0h	7h
Home tests	0.5h	0h	0.5h
Farm tests	0h	7h	7h
Ask for help tests	0h	0.5h	0.5h
Ranking tests	1h	0h	1h
Report tests	0h	1h	1h
Forum tests	2h	0h	2h
Weather forecast tests	0.5h	0.5h	1h
	87.5h	87.5h	177h

References

¹ Specification Document: "02. Assignment IT AY 2021-2022.pdf"

² Slides of the lectures