POLITECNICO
MILANO 1863

DREAM project by Federica Tommasini & Alessandro Verosimile

Acceptance Testing Document

Deliverable:	ATD
Title:	Acceptance Testing Document
Authors:	Federica Tommasini, Alessandro Verosimile
Version:	1.0
Date:	11-February-2022
Download page:	https://github.com/federicatommasini/TommasiniVerosimile.git
Copyright:	Copyright © 2022, Federica Tommasini and Alessandro Verosimile – All rights reserved

Contents

Table of Contents **3**

List of Figures **4**

List of Tables **4**

1 Tested project **5**
1.1 Authors 5
1.2 Repository URL 5
1.3 Documents considered 5

2 Installation **5**

3 Project inspection **6**
3.1 Acceptance test cases 6
3.1.1 Authentication 6
3.1.2 Farmer functionalities 7
3.1.3 Policy maker functionalities 11

4 Conclusions **12**

5 Effort Spent **13**

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

List of Figures

1 The profile of a policy maker shows wrong information. 7
2 Empty suggestions section 8
3 Help request non received 10

List of Tables

1 Test case 1 6
2 Test case 2 7
3 Test case 3 8
4 Test case 4 8
5 Test case 5 9
6 Test case 6 9
7 Test case 7 10
8 Test case 8 11
9 Test case 9 12

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

1 Tested project

1.1 Authors

- Kinga Marek
- Józek Piechaczek
- Mariusz Wiśniewski

1.2 Repository URL

https://github.com/Nexer8/MarekPiechaczekWisniewski.git

1.3 Documents considered

- RASD: Requirements Analysis Specification Document
- DD: Design Document
- ITD: Implementation and Testing Document

2 Installation

In order to run and test the application, it has been followed the installation guide. It has been decided to run the application utilizing Docker Desktop, therefore it has been previously installed the recommended version through the link provided in the guide. After installing Docker, the zip folder containing all the code has been downloaded from the GitHub repository and, as indicated in the installation guide, it has been executed the following command in the code directory:

docker - compose up -d –build

Once the installation process has been terminated, it has been started the testing of the application, running the client into a browser using Docker.

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

3 Project inspection

In order to test the functioning of the application, there have been defined some acceptance test cases, that are aimed to verify the satisfaction of the requirements that have been chosen to implement and that have been specified into the ITD document.

3.1 Acceptance test cases

3.1.1 Authentication

The following test cases are aimed to verified the correct functioning of the authentication of the users of the application. For each test case are specified the requirements to which is mapped and the result that is obtained. Before proceeding with the acceptance test, it has been noticed that every user can create an account with the role of a policy maker without any security check. There is not any requirement that specifies this, so it is coherent with them. However, it is not a good choice to allow everyone to create a profile as a policy maker, an user who can manage the evaluations of every farmer and take a lot of relevant decisions.

User registration		
Description	**Requirements**	**Result**
The goal is to verify if the registration process works accordingly to what is specified in the requirements. From the presentation page of the application, clicking on the "create account" button, it appears a form for the registration, which contains several required fields regarding the information about the farmer and its farm if the chosen role is "farmer" or about the policy maker if the chosen role is "policy maker". All the fields are mandatory. If all the data are correctly inserted, the system creates the account; if the user tries to insert an email which is already registered he gets an alert and has to insert a different one in order to complete the process.	• R1: The system must uniquely identify each user by his e-mail. • R2: The system must allow an unregistered user to create an account with a chosen role. • R4: The system must ensure that a farmer inserts his farm data during the registration process.	Passed

Table 1: Test case 1

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

User login		
Description	**Requirements**	**Result**
The goal is to verify if the login process works accordingly to what is specified in the requirements. From the presentation page of the application, clicking on the "Log in" button, it appears a popup that requires email and password; if the user inserts them correctly, the system allows him to log into the application. Otherwise, the user gets an alert. Once the user is logged in, he can visualize all his personal information and access to all the functionalities. He also can sign out from the application clicking the "Sign out" button and deleting his account clicking the "delete account" button; several checks about the proper work of these functionalities are performed: for example, it has been tried to log in with the same credentials of an account that has been deleted. Only one bug has been found in this test: for what concerns the policy maker, whatever credentials are used to log into the application, the name, surname and email reported in the personal information are Assish Rai, pmaker@pmaker.it	• R6: The system must allow a registered user to log into the application. • R8: The system must allow a logged-in user to sign out from the application. • R9: The system must allow a registered user to delete his account.	Partially passed

Table 2: Test case 2

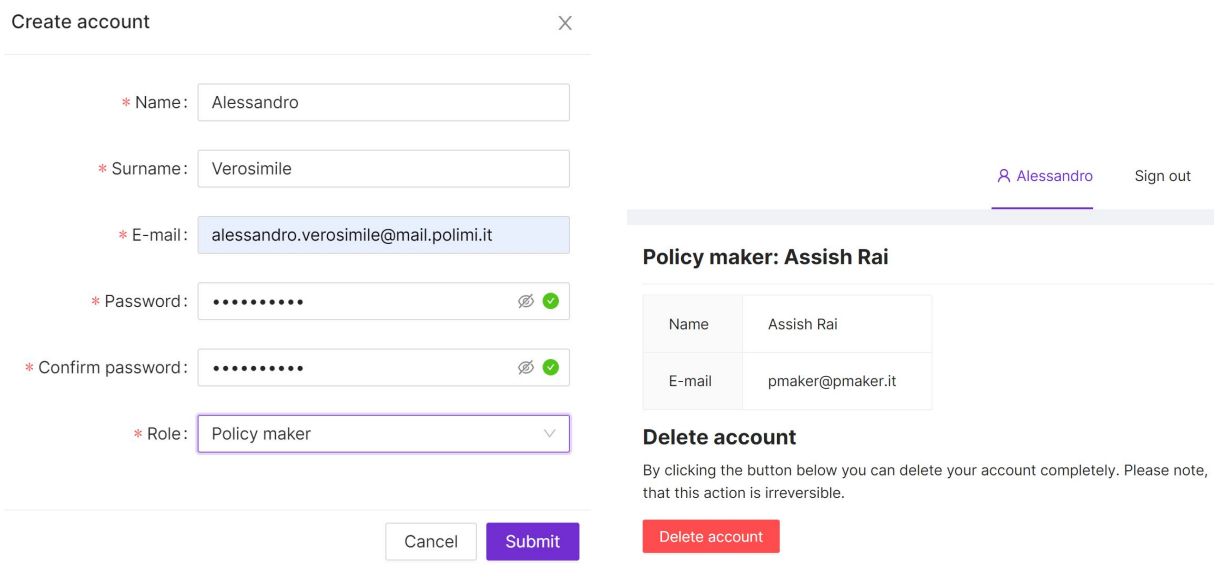

Figure 1: The profile of a policy maker shows wrong information.

3.1.2 Farmer functionalities

The following test cases are aimed to verified the correct functioning of the farmer side of the application. For each test case are specified the requirements to which is mapped and the result that is obtained.

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

Farmer home section		
Description	**Requirements**	**Result**
The goal is to verify if some of the main functionalities of the farmer side of the application work properly. A logged-in farmer, from the home page of the application can manage and visualize some information. In particular he can visualize his personal summary, all the help requests he uploaded and manage his monthly production. For what concerns the personal summary and the help requests, they work properly; they are tested with several changes to the information in order to verify if they change in a coherent way. Also for what concerns the management of monthly production, all works properly.	• R24: The system must allow a farmer to see his own farmer's summary. • R26: The system must allow a farmer to manage his monthly production data. • R27: The system must allow a farmer to see a list of help requests created by him.	Passed

Table 3: Test case 3

Farmer suggestions		
Description	**Requirements**	**Result**
The goal is to verify if the visualization of the suggestions in the home page of the application work properly. In the home page a farmer can visualize personalized suggestions but this functionality does not work very well. Their visualization is rather ambiguous: it is not clear if they are uploaded by agronomists or other farmers; it is possible to guess it by R46 in which it is written that agronomists manage a list of suggestions (but does "manage" mean that they write them?). Anyway this list in the home page is sometimes empty and sometimes filled with something but it is not possible to understand how to test the correct visualization of it.	• R25: The system must allow a farmer to see personalized suggestions.	Not passed

Table 4: Test case 4

Figure 2: Empty suggestions section

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

Farmer help request section		
Description	**Requirements**	**Result**
The goal is to verify if the functionalities of the farmer side of the application concerning the help request section work properly. A logged-in farmer, from the help request section of the application can manage and visualize his help requests. In particular he can create a new help request, search them by the topic and delete them. With several attempts the proper work of all these functionalities has been tested and all is coherent.	• R28: The system must allow a farmer to find a help request created by him by the topic. • R29: The system must allow a farmer to see a specific help request created by him. • R30: The system must allow a farmer to delete a specific help request created by him. • R31: The system must allow a farmer to create a new help request.	Passed

Table 5: Test case 5

Farmer forum section		
Description	**Requirements**	**Result**
The goal is to verify if the functionalities of the farmer side of the application concerning the forum section work properly. A logged-in farmer, from the forum section of the application can manage and visualize forum threads. In particular he can visualize all the available forum threads, find a specific one by the topic, create a new one, visualize a specific one and write and delete comments into it. With several attempts such as the creation of various forum threads, the insertion of new comments and the deletion of them, the proper work of all these functionalities has been tested and all is coherent.	• R38: The system must allow a farmer to see a list of all forum threads in Telengana. • R39: The system must allow a farmer to find a forum thread by the topic. • R40: The system must allow a farmer to see a specific forum thread with its comments. • R41: The system must allow a farmer to create a comment in a forum thread. • R42: The system must allow a farmer to delete a comment in a forum thread, only if it was created by him. • R43: The system must allow a farmer to create a forum thread.	Passed

Table 6: Test case 6

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

Farmer provide help section		
Description	**Requirements**	**Result**
The goal is to verify if the functionalities of the farmer side of the application concerning the provide help section work properly. A logged-in farmer, positively evaluated by a policy maker, from the provide help section of the application can visualize and answer to help requests of other farmers. Firstly, it has been tested if this functionality is effectively present only for the farmers who are positively evaluated and the test has been passed. Then all the relative functionalities have been tested. In this section a farmer can see a list of all help requests of his mandal, to find a specific one by the topic, to respond to a specific one and to delete a specific response if it has been uploaded by him. Finally he can visualize the summaries of the farmers whose help requests he received. For what concerns the visualization of the help requests it does not work completely well: if a farmer becomes positively evaluated he cannot visualize the help requests previously published but only the ones published from the moment in which he becomes positively evaluated. For what concerns the other functionalities, they have been tested with several attempts, such as the creation of help requests by farmer of different mandals and the visualization of them by farmers of the same mandal or different mandal, the visualization of the summaries and so on, and all is coherent.	• R32: The system must allow a farmer with a positive note to see a list of all help requests in his mandal. • R33: The system must allow a farmer with a positive note to find a help request in his mandal by the topic. • R34: The system must allow a farmer with a positive note to see a specific help request in his mandal. • R35: The system must allow a farmer with a positive note to respond to a specific help request in his mandal. • R36: The system must allow a farmer with a positive note to delete a help response, only if it was created by him. • R37: The farmer is able to see farmer's summary of a farmer with a negative note whose help request he received.	Partially passed

Table 7: Test case 7

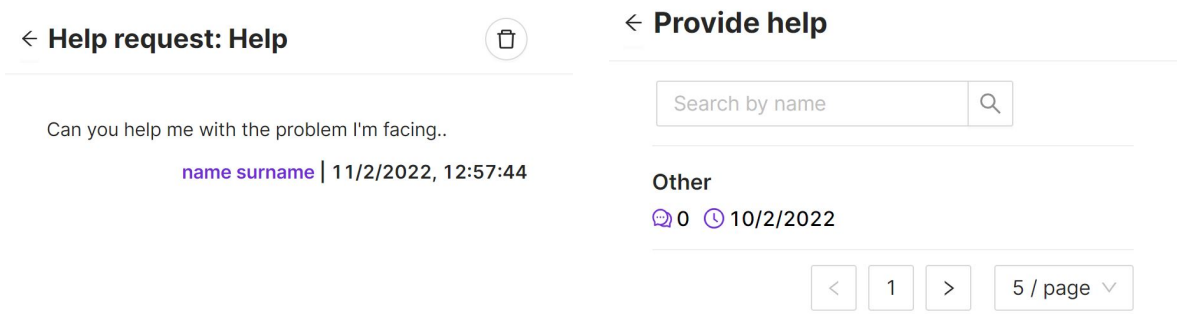

Figure 3: Help request non received

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

3.1.3 Policy maker functionalities

The following test cases are aimed to verified the correct functioning of the policy maker side of the application. For each test case are specified the requirements to which is mapped and the result that is obtained.

Visualization of farmers' information		
Description	**Requirements**	**Result**
The goal is to verify if the functionalities regarding the visualization of the farmers' information by a policy maker work accordingly to what is specified in the requirements. After logging in as a policy maker, it has been tested that, going to the farmers section, it is possible to visualize the list of all the farmers in Telengana. In addition it has been verified that such list can be filtered with respect to mandal or to the type of note associated to a farmer. Furthermore, it has been verified that a specific farmer can be found by writing his name in the search bar. Finally, for each farmer, it is visible an icon that leads the policy maker to visualize the summary with the information regarding the selected farmer. Therefore, all the functionalities described in the requirements work as expected.	• R19: The system must allow a policy maker to see a list of all farmers in Telangana. • R20: The system must allow a policy maker to see a list of all farmers with a specific note. • R21: The system must allow a policy maker to see a list of all farmers in a given mandal. • R22: The system must allow a policy maker to find a farmer's summary by his name and surname. • R23: The system must allow a policy maker to see a farmer's summary.	Passed

Table 8: Test case 8

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

Assignment of a note to a farmer		
Description	**Requirements**	**Result**
The goal is to verify if the functionalities regarding the assignment of a note to a farmer by a policy maker work accordingly to what is specified in the requirements. After having visualized the summary of a specific farmer, it has been tried to change the note of a farmer. It has been observed that if the policy maker changes it to a negative note, he is required to specify a problem type choosing it from a list of possible ones, while, if the updated note is neutral or positive, no additional information are needed. It has been also checked that, registering a new farmer in the application, his note it's initially set as neutral. Finally, it has been checked that after changing a note of a farmer to a negative one, an help request is automatically generated. All the above described tests provided the expected results.	• R10: The system must allow a policy maker to assign a note to a farmer. • R11: The system must ensure that every farmer initially has a neutral note. • R12: The system must have a list of predefined farmer's problem types. • R13: The system must allow a policy maker to specify a problem type when assigning a negative note. • R17: The system must ensure that an automatic help request is created, if a farmer receives a negative note.	Passed

Table 9: Test case 9

4 Conclusions

In conclusion, the application has been realized in a coherent way. Most of the requirements are precisely respected, some of them not completely. However, in general the requirements are not easy to comprehend, often are really ambiguous and, for what concerns some of them, it is needed some time to understand what they are referred to. The application has got the requested functionalities but in some sections it is not really intuitive to understand how some of them are carried out.

However, considering that the application is a prototype, it is mostly coherent and quite well implemented.

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved

5 Effort Spent

Student	Hours
Student 1	10h
Student 2	10h

Copyright © 2022, Federica Tommasini, Alessandro Verosimile – All rights reserved