

DREAM

Data-driven Predictive Farming in Telengana

*Requirement Analysis and Specification
Document - RASD*

Christian Grasso - Filippo Lazzati - Chiara Magri
Year: 2021/2022

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, Acronyms, Abbreviations	2
1.4	Revision history	2
1.5	Reference Documents	2
1.6	Document Structure	2
2	Overall description	2
2.1	Product perspective	2
2.2	Product functions	2
2.2.1	Data collection	2
2.2.2	Data analysis	3
2.2.3	Forum	3
2.2.4	Request and supply of help	3
2.2.5	Daily plan	3
2.3	User characteristics	3
2.4	Assumptions, dependencies and constraints	4
3	Specific requirements	4
3.1	External Interface Requirements	4
3.2	Functional Requirements	4
3.3	Performance Requirements	4
3.4	Design Constraints	4
3.5	Software System Attributes	4
4	Formal analysis using Alloy	4
5	Effort spent	4
6	References	4

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

1.4 Revision history

1.5 Reference Documents

1.6 Document Structure

2 Overall description

2.1 Product perspective

2.2 Product functions

The main goal that Telengana's government wants to achieve with **DREAM** is to help policy makers formulating policies in the field of agriculture. In order to accomplish this objective, Telengana's government is asking for predictive models for food systems that can drive decisions exploiting huge amount of data. In this section, a list of the most important requirements of the system is provided; notice that they are just briefly described, since they will be analysed in-depth in chapter 3.

2.2.1 Data collection

Just like every data-driven decision, data is essential. **DREAM** must be able to manage different kinds of data coming from different sources:

1. agronomists and farmers insert into the system standard kind of information about the farms they respectively visit and own, but also other kinds of information like, for example, the types of products cultivated and the production volume for each product (kinds of information that allow the analytics to profile these users). **DREAM** also allows them to insert less-structured data, for instance full-text feedback about the inserted data (farmers can insert data about any problem they face);
2. data gathered by sensors deployed on the territory and by the water irrigation system are automatically collected by **DREAM** through effective interfaces interacting with such systems;
3. already existing systems are integrated with **DREAM**;
4. the last but not the least, farmers can interact through discussion forums, that are stored by **DREAM**.

2.2.2 Data analysis

The raw data collected by DREAM must be processed before being delivered to the end-user. Therefore, starting with the big volume of information “ingested”, various kinds of analytics are performed in order to provide a more aggregate version of the data:

- the suggestions provided to the farmers are customized according to their information;
- information about farmers’ production is used to distinguish among farmers who are performing well and those who are not;
- patterns and relations are identified among the data (for instance between weather forecasts and production volumes);
- analytics are also used for quantifying the improvement (if present) in the crop after having adopted agronomists’ suggestions.

2.2.3 Forum

As previously mentioned, farmers are allowed to open discussion forums on DREAM with other farmers. This allows to keep in touch with other people doing the same job and, hence, to ask for advice in case of needing.

2.2.4 Request and supply of help

There are 4 different ways for a farmer to ask for help: ask for it directly to other farmers, ask in the forum, ask to the agronomists or waiting for the system to recognize him as in trouble. As far as the first option is concerned, the service is provided through a GUI. On the other side, either farmers who are performing well or agronomists may be reached by the system on behalf of other farmers (or of the system itself) for a request of help. Then the system introduces them. It should be remarked that a farmer can contact only agronomists in the same area.

2.2.5 Daily plan

DREAM automatically devises a daily plan for each agronomist consisting in a list of farms that must be visited and provides a possible time schedule for such meetings. Every daily plan must be actively accepted by the agronomist before the involved farmer is notified. Anyway, the agronomist can apply some changes to the schedule, but the system will not approve them if some constraints are violated.

2.3 User characteristics

With regards to the possible actors of DREAM, three different main user classes can be identified:

Policy makers: their job is to devise policies to regulate the agricultural production. DREAM helps them providing aggregate data. Policy makers access the system and then receive high-level information about which farmers are performing well and which not, and possible reasons to this fact. Furthermore, DREAM helps to understand the impact of agronomists' clues in farmers' productions;

Farmers: they have a farm and some plots of land to cultivate. They access the system and insert data about themselves, helping DREAM to profile them. They insert data about their productions, the area where they work, eventual problems they must face, questions in the forum ... and so on. On the other side, DREAM puts them in touch with someone who can help them and provides data that may be relevant to them;

Agronomists: an agronomist has to find methods for increasing the production and the quality of the harvest. They insert the area they are responsible of and they receive data about the best performing farmers and weather forecasts. They can answer to farmers' questions and they receive a raw daily plan about which farmers they should visit.

2.4 Assumptions, dependencies and constraints

3 Specific requirements

3.1 External Interface Requirements

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

4 Formal analysis using Alloy

5 Effort spent

6 References