SWAMP

SMART WATER MANAGEMENT PLATFORM

Project nº: 777112

WP4

D4.2 Smart Precision Irrigation Application

(Version 1.0)

Editor: Ramide Dantas (UFPE)

Author(s): Ramide Dantas (UFPE)

Milton V. da Gama Neto (UFPE)

Status – Version: 1.2

Date: Nov 2019

Distribution – Confidentiality: Public

Code: 777112-SWAMP - D4.2 Smart Precision Irrigation Application

Disclaimer

Document revision history

Date	Issue	Author/Editor/Contributor	Summary of main changes
25 September 2019	0.0	Ramide Dantas (UFPE)	Preliminary structure
15 October 2019	0.1	Ramide Dantas (UFPE)	Structure changes; base content: stories, API, screens.
25 October 2019	0.2	Ramide Dantas (UFPE)	Added text to main sections.
05 November 2019	0.3	Ramide Dantas (UFPE)	Added text; revised structure; API section revised;
06 November 2019	1.0	Ramide Dantas (UFPE)	Added new screenshots, download link.
20 November 2019	1.1	Ramide Dantas (UFPE)	Internal review fixes.
21 November 2019	1.2	Ramide Dantas (UFPE)	Final clean-up for submission.

Internal review history

Date	Reviewer	Summary of comments
11 November 2019	Hannu Tanner (VTT)	Document approved with some minor questions and remarks on content.
18 November 2019	Ronaldo Prati (UFABC)	Minor typos and revision

Table of contents

Abbreviations 5
Executive Summary 6
1. Introduction 7
1.1. Overview 7
1.2. Deliverable structure 7
2. User Stories 7
3. Architecture and Technologies 9
3.1. SWAMP API 9
4. Main Application Screens 12
5. Final Considerations and Next Steps 15
6. References 15

Abbreviations

AAPT	Android Asset Packaging Tool
ADB	Android Debug Bridge
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
REST	Representational state transfer

Executive Summary

The present deliverable D4.2 describes the version 1.0 of the Farmer Application for Smart Irrigation of the SWAMP project. It lists the functionalities that were implemented as well as those planned for the next implementation cycle (due month 30). The functionalities follow to the specification provided in the interim report of the Task 4.2 [1]. Additional details on the implementation are provided, including the development of an API used by the application. A comparison of the original specification in [1] and the actual screens as implemented is provided.

1. Introduction

1.1. Overview

This report describes the Smart Precision Irrigation Application developed by the SWAMP Project, also called "Farmer App" in the remaining of this document and other project reports. As the name suggests, the application is intended for the decision-maker at the farm, who will monitor the water status of the fields and ultimately decide whether and how much to irrigate, with the help of the estimates, forecasts, and plans provided by the platform. In order to allow the farmer to see and control his/her farm anytime and anywhere, the project opted for a mobile application for smartphones/tablets. Platform configuration and maintenance functionalities are not covered in the application since its intent is helping the farmer with irrigation activities. These functionalities, however, are available through a Web-based management interface, including the instantiation of Virtual Entities (Farm, Fields, etc.), configuration and monitoring of sensors, actuators, drones, etc. This management UI is intended for system integrators, administrators, IoT professionals and the like, and is described in section 5 of the Deliverable 1.3 Deployment and Management Services [3].

1.2. Deliverable structure

This document is organized as follows:

- Section 2 describes the User Stories as gathered from pilot users, indicating the implementation status of each one;
- Section 3 brings more details on the application implementation, including the technologies used and an API specification for the platform;
- Section 4 shows the User Interface of the application as of Version 1.0, comparing it with the specification provided at M18;
- Section 5 contains the final considerations and next steps.

2. User Stories

Table 1 lists the User Stories as described in the Interim Report for the Farm prototype [1] and their implementation status in the version 1.0 of the Farmer Application. Priority was given to features that allowed the farmer to monitor the water status of the farm as well as accessing and applying the irrigation plan, since they are seen as critical by farmers. The remaining features will be added incrementally, being planned for the updated version at month 30. New features, improvements and corrections are also expected as the application receives feedback from the use in the pilots. Table 1 will be updated accordingly and the priorities revised for the upcoming report.

TABLE 1 – USER STORIES FOR THE FARMER APPLICATION

SCENARIO 1: MONITORING WATER STATUS			
Identifier	Story	Priority	Status
1.1	"As a Farmer, I want to see the general water status of my farm – including a forecast of water need, precipitation and irrigation – , so that I can make decisions about irrigation operations."	Critical	Partial (v1)
1.2	"As a Farmer, I want to view the water status of individual fields, both current and past, in order to best understand the water dynamics of the fields and intervene if necessary."	Critical	Implemented (v1)
1.3	"As a Farmer, I want to be notified in case any field's water level is below pre-established thresholds, in order to take corrective action by activating the irrigation systems."	Important	Planned for update (v2)
SCENARIO 2: REQUESTING IRRIGATION PLANS			
Identifier	Story	Priority	Status
2.1	"As a Farmer, I want to send my water demand request for the next few days to the water distribution company and receive a reply in the application."	Important	Planned for update (v2)
2.2	"As a Farmer, I want the application to provide me with an Irrigation Plan that minimizes my operational cost (e.g., energy) while avoiding water stress by the crops."	Critical	Implemented (v1)
3.3	"As a Farmer, I want the application to provide me with statistics that can help me improve the irrigation system of my farm."	Moderate	Planned for the future
SCENARIO 3: ADJUSTING/CREATING IRRIGATION PLANS			
Identifier	Story	Priority	Status
3.1	"As a Farmer, I want to adjust the Irrigation Plan computed by the application to fit my expertise in crop management."	Important	Planned for update (v2)
3.2	"As a Farmer, I want to create my own Irrigation Plan according to my experience."	Important	Planned for update (v2)
SCENARIO 4: APPLYING AND MONITORING IRRIGATION			
Identifier	Story	Priority	Status
4.1	"As a Farmer, I want to execute an Irrigation Plan in the farm via the automated irrigation systems in place."	Critical	Implemented (v1)
4.2	"As a Farmer, I want to monitor the execution of an Irrigation Plan, in order to know the current status and verify it in the field."	Important	Planned for update (v2)

SCENARIO 5: DRONE AND REMOTE SENSING			
Identifier	Story	Priority	Status
5.1	"As a Farmer, I want to able to visualize Remote Sensing (Satellite and Drone) imagery in order to help in my decision making."	Important	Planned for update (v2)
5.2	"As a Farmer, I want to request Drone flights over specific fields through the Application."	Moderate	Planned for update (v2)

3. Architecture and Technologies

The Farmer Application was implemented as a native Android application, using standard Android tools (Android Studio, AAPT, ADB, etc.). The choice for Android was due to its market dominance (85% world-wide[1] and Brazil[2], and more than 70% in Europe[3]). Although the Android platform is migrating to the Kotlin programming language, the application was developed using Java, which is more widely known and supported. The application communicates with SWAMP platform through a REST API specified and developed for this purpose. An overview of this interaction is shown in Figure 1.

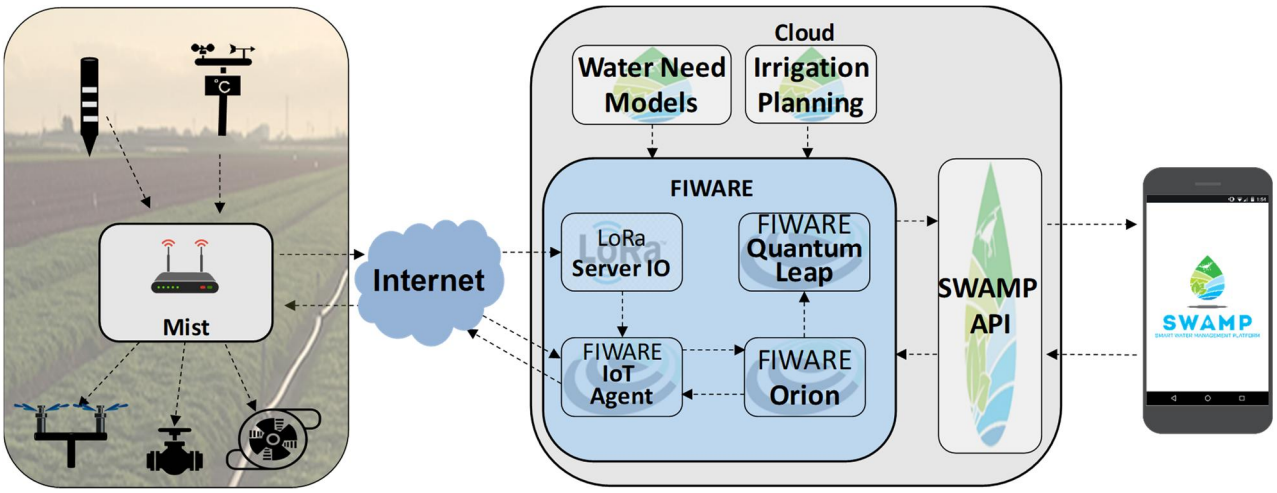


FIGURE 1 – INTERACTION OF THE FARMER APP WITH THE SWAMP PLATFORM.

3.1. SWAMP API

The SWAMP API provides a unified entry point for application to access the platform services. It facilitates the implementation of new applications, offering a richer alternative to the services provided by the FIWARE platform. It also offers additional security and notification capabilities, which are critical for modern,

[1] https://www.idc.com/promo/smartphone-market-share/os

[2] https://gs.statcounter.com/os-market-share/mobile/brazil

[3] https://www.statista.com/statistics/639928/market-share-mobile-operating-systems-eu/

functional mobile applications.

The API follows a REST philosophy, using HTTP and JSON as communication protocol and data format, for universal support. Table 2 describes the API endpoints for the current implementation of version 0 of the API, which is a draft specification intended for incremental evolution as the farmer application itself matures. The Entity types are those described in the Deliverable 2.1 - Communication and Storage Substrate [2]. The return format is a JSON list of the historical records for a given entity (or entities), using the Quantum Leap (FIWARE Generic Enabler) format, and is illustrated in FIGURE 2 (manually formatted and some fields removed for clarity). This JSON shows the last ten records for the Probe named **`urn:ngsd-ld:ProbeGuaspari:2`**, including three different moisture readings (for different depths).

TABLE 2 - SWAMP API SPECIFICATION (VERSION 0) FOR THE FARMER APP.

URL	Result
`/v0/<Entity Type>/<ID>`	Returns all attributes for the last 10 records of the entity identified by ID
`/v0/<Entity Type>/<ID>?limit=n`	Returns all attributes for the last N records of the entity identified by ID
`/v0/<Entity Type>`	Returns all attributes for the last 10 records for all entities of a given type
`/v0/<Entity Type>?limit=n`	Returns all attributes for the last N records for all entities of a given type
`/v0/<Entity Type>/<ID>?<attribute>`	Returns a given attribute for the last 10 records of the entity identified by ID
`/v0/<Entity Type>/<ID>?<attribute>?limit=n`	Returns a given attribute for the last N records of the entity identified by ID
`/v0/<Entity Type>/<ID>?<attribute>=<value>`	Returns all attributes for all records that match <attribute>=<value> of the entity identified by ID
`/v0/<Entity Type>?<attribute>=<value>`	Returns all attributes for all records that match <attribute>=<value> for all instances of that type.

```
[
   {
      "attributes":[
           {
            "attrName":"identifier",
            "values":[ 1002, 1002, 1002, 1002, 1002, 1002, 1002, 1002, 1002, 1002]
           },
           {
            "attrName":"moisture1",
            "values":[ 385, 366, 360, 281, 281, 281, 280, 279, 278, 278]
         },
         {
            "attrName":"moisture2",
            "values":[ 621, 604, 596, 462, 461, 460, 458, 454, 453, 452]
         },
         {
            "attrName":"moisture3",
            "values":[ 613, 611, 611, 626, 626, 625, 625, 624, 624, 624]
         }
      ],
      "entityId":"urn:ngsd-ld:ProbeGuaspari:2",
      "index":[
         "2019-08-02T13:00:04.344",
         "2019-08-02T14:00:04.390",
         "2019-08-02T15:20:04.418",
         "2019-10-03T11:10:05.582",
         "2019-10-03T11:20:07.631",
         "2019-10-03T11:30:05.403",
         "2019-10-03T11:50:05.571",
         "2019-10-03T12:40:05.433",
         "2019-10-03T12:50:05.442",
         "2019-10-03T13:00:05.431"
      ]
   }
]
```


FIGURE 2 – EXAMPLE OF JSON RETURNED BY THE SWAMP API.

4. Main Application Screens

This section presents a side-by-side comparison of the screens as described in the interim application report [1] and the version 1.0 of the application as currently implemented. Figure 3 shows the navigation flow of the user through the main screens of the application, as specified in [1]. This section assumes that the application and platform have been installed and configured successfully, and the data regarding the Farm, Crop, Irrigation system, etc., has been successfully entered into the system.

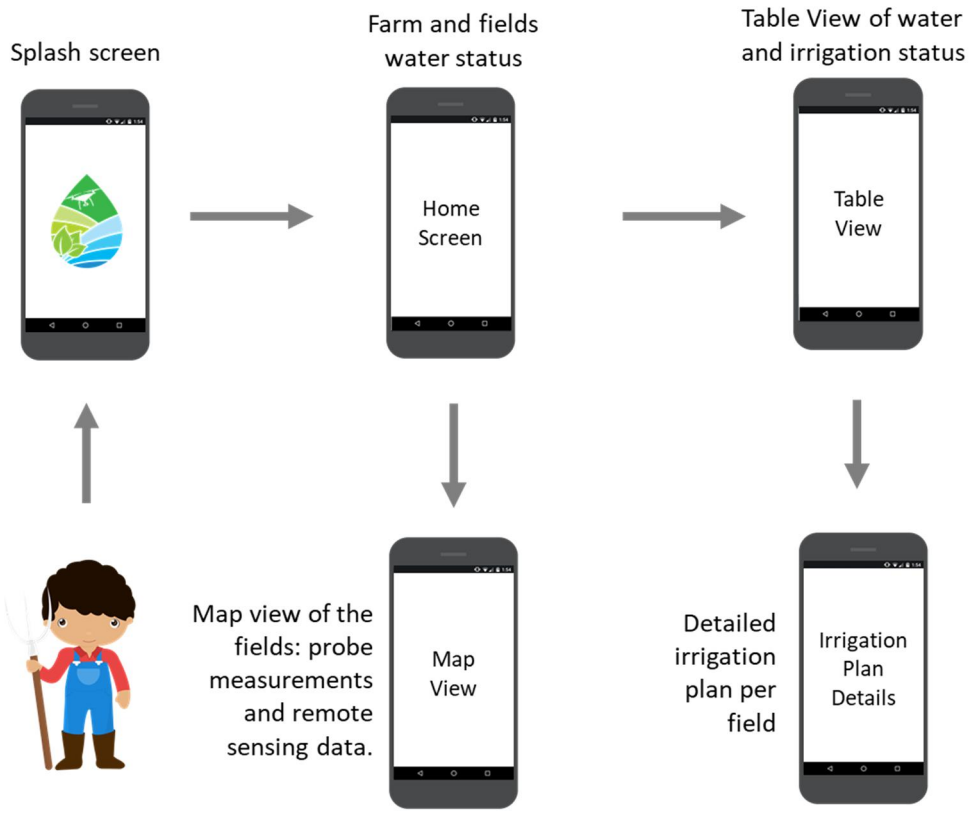
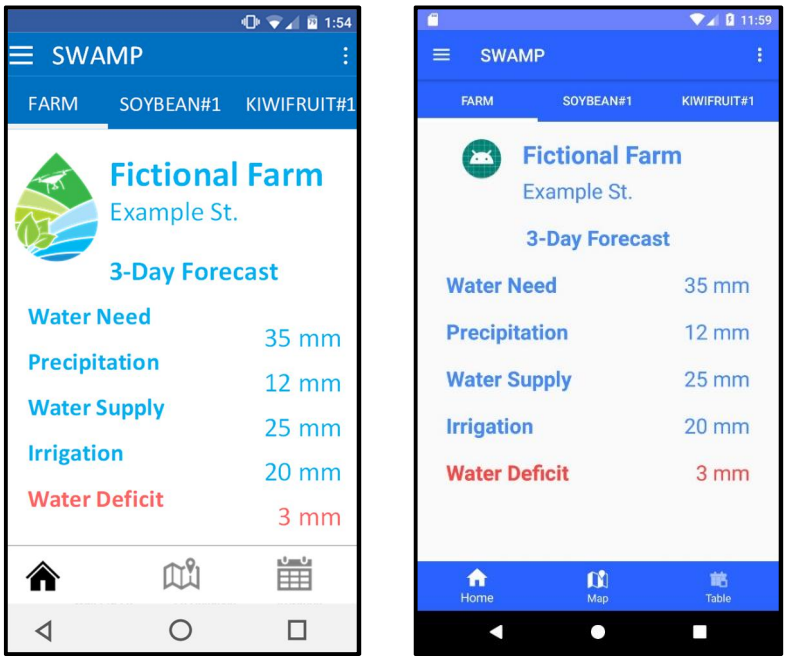


FIGURE 3 – NAVIGATION SCHEME OF MAIN SCREENS.

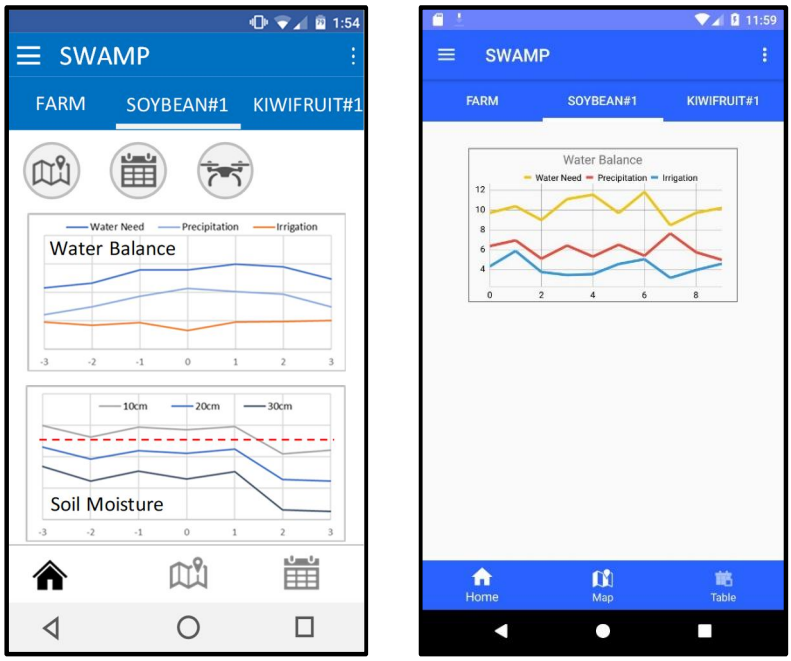
In Figure 4, it can be seen that the home screen of the application (Figure 4b) follows the original specification (Figure 4a) closely. Its goal is to provide a quick summary of the water status of the farm, as to fulfil user story 1.1 (General Farm Status).



a) Original Design b) Implementation

FIGURE 4 – FARMER APPLICATION HOME SCREEN.

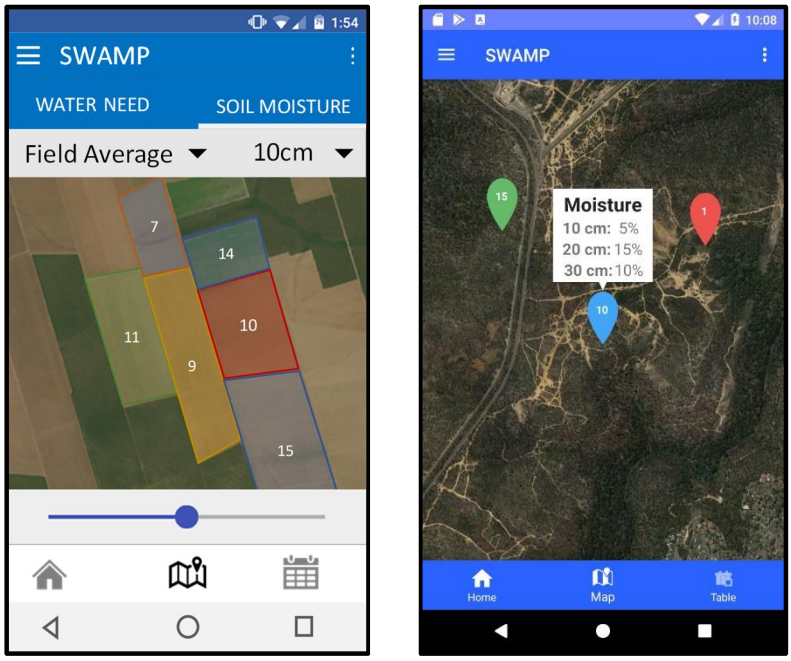
The water status for individual fields is shown in Figure 5, including charts with historical data, addressing user story 1.2 (Individual Field Status). The current version (Figure 5b) does not include the shortcuts yet to other screens and for drone activation (round buttons at the top of Figure 5a). Through this screen, the farmer will be able to visualize the soil moisture notification threshold (user story 1.3), although this functionality is not implemented in this version.



a) Original Design b) Implementation

FIGURE 5 – FIELD WATER STATUS SCREEN.

The satellite view of the fields is shown in Figure 6. In the specification (Figure 6a), this view shows both water need estimates and current soil moisture. A “time slider” at the bottom allows navigating backward for visualizing past data, as well as forward for future estimates. The current implementation (Figure 6b) is a simplified version of the specification, showing a map with markers for the moisture probes in the fields. Marker colour is used to represent the moisture level (e.g., blue as high, red as low, and so on). When clicked/touched, markers provide moisture readings for the various depths in the probe. As implemented, this view should complement user story 1.2 (field monitoring) and be a starting point for story 5.1 (visualization of remote sensing data), planned for the next iteration.



a) Original Design b) Implementation

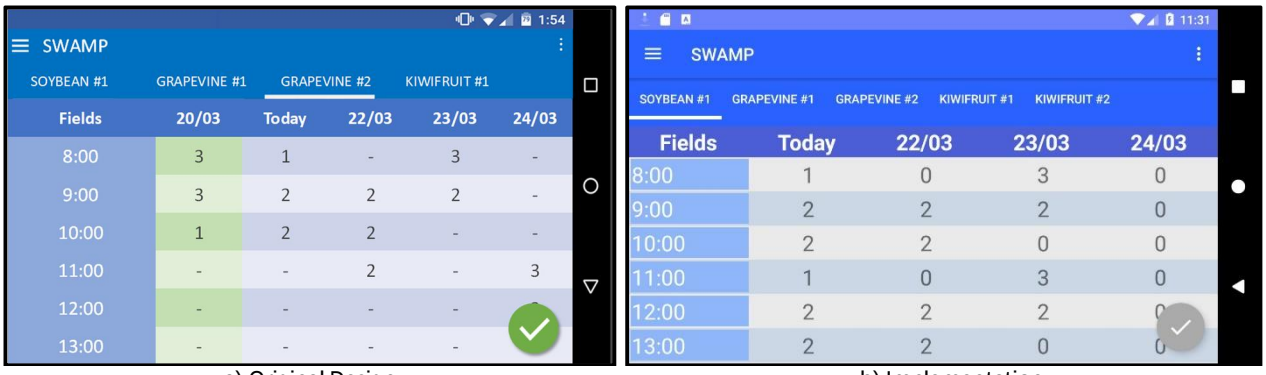
FIGURE 6 – MAP VIEW WITH SENSOR DATA.

The screen in Figure 7 presents the farmer with numeric estimates in a tabular fashion for all the fields. This screen address user stories 1.2 (field monitoring) and 2.2 (irrigation plan). As of the time of implementation, irrigation plans are computed automatically by the platform as new estimates are available (Water Need, weather forecast, and so on), and therefore, the farmer does not need to request them via the application.

a) Original Design b) Implementation

FIGURE 7 – TABLE VIEW WITH WATER NEED ESTIMATES.

The detailed irrigation plan for a given field is shown in Figure 8. In these screens, it is possible to visualize the amount of water that will be irrigated at each moment. Through this screen, the farmer can also apply the irrigation plan, which changes its status in the platform and triggers the irrigation procedures (story 4.1).



a) Original Design b) Implementation

FIGURE 8 – TABLE VIEW WITH DETAILED IRRIGATION PLAN FOR A FIELD.

5. Final Considerations and Next Steps

This report describes the version 1.0 of the Farmer Application for Smart Irrigation of the SWAMP Project. The present version of the App focuses on basic monitoring functionality as well as irrigation activation. The intelligence behind these functionalities resides in the SWAMP Platform itself, in the form of services running in the cloud. The platform and its services are accessed through the SWAMP API for Water Management, specified and developed alongside the Farmer Application. The API protects inner components of the platform from malicious access and facilitates Application development – and therefore exploitation – of the platform.

The role of the App is to provide the farmer with easy access to the platform smart services, and for that purpose, several rounds of tests in the field will be necessary. New functionalities and other improvements are expected as feedback from end-users is gathered. These will be implemented along with the remaining functionalities described in section 2. Improvements in the SWAMP API are also planned, as the functionalities in both the platform and the App mature.

6. References

[1] Dantas R. (ed.) et al., “Smart Precision Irrigation Application”, SWAMP Deliverable D4.2 (interim report), prototype version, April 2019.

[2] Filev, R. (ed.) et al., “Communication and Storage Substrate”, SWAMP Deliverable D2.1, November 2019.

[3] Kamienski, C. (ed.) et al., “Deployment and Management Services”, SWAMP Deliverable 1.3, April 2019.