



CTIS-411 SENIOR PROJECT-I

SOFTWARE REQUIREMENTS SPECIFICATION
(SRS)



Team 5 - Members

Berk Özdoruk
Erdoğan Yağız Şahin
Oğuzhan Özkan
Ömer Levent Durdalı

Project Supervisor

Dr. Cüneyt Sevgi

CTIS

COMPUTER TECHNOLOGY AND
INFORMATION SYSTEMS

Table of Contents

Requirements	3
Functional Requirements	3
Use Case Diagram	3
Use Case Narratives	5
Non-Functional Requirements :	33
Constraints:	33
Requirements Prototype:	34
System Models:	35
Activity Diagrams:	35
System Sequence Diagrams:	37
Analysis Class Diagram:	39

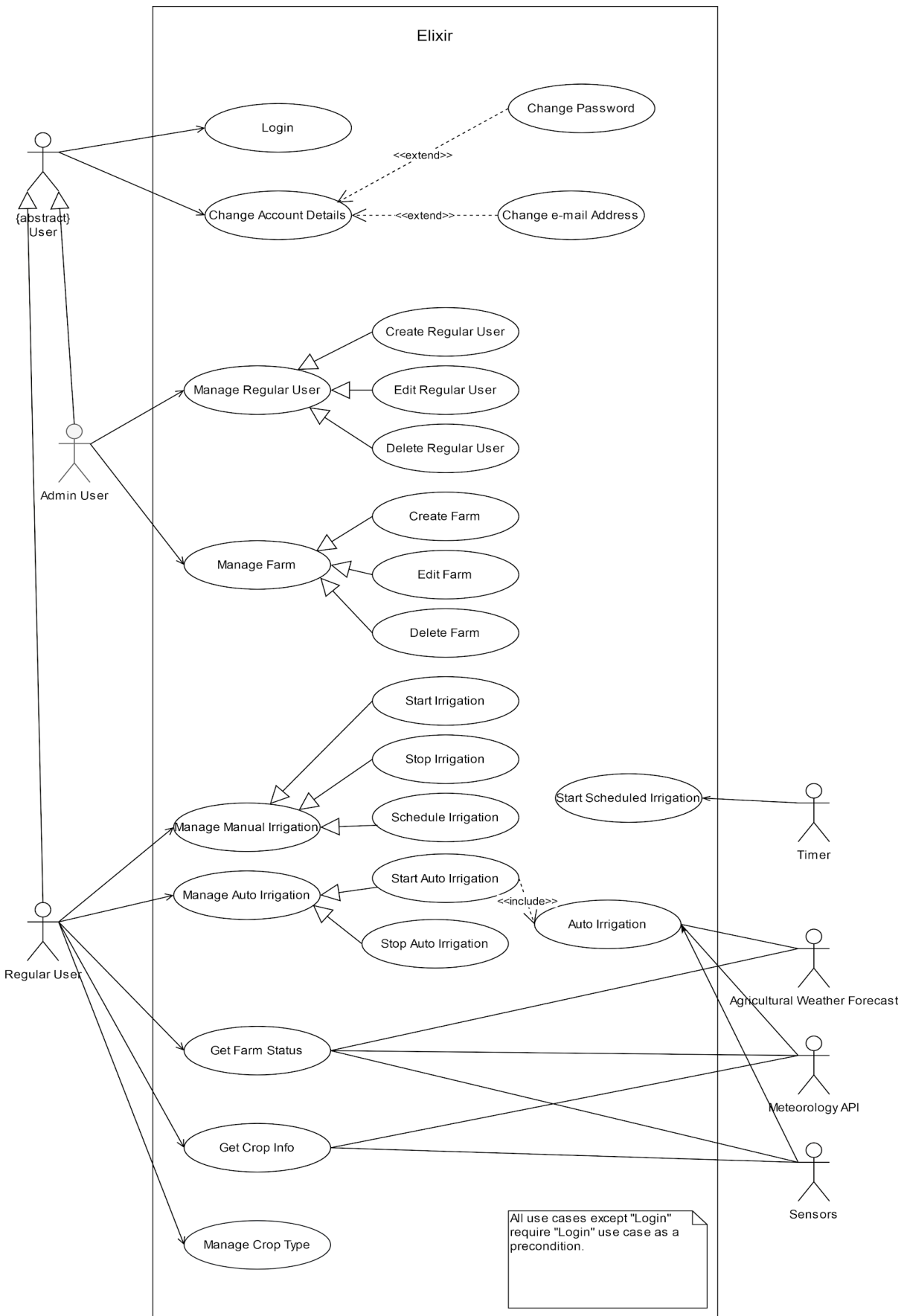
List of Figures

Figure 1: Elixir Use Case Diagram	4
Figure 2: Get Farm Status Requirements Prototype	34
Figure 3: Manage Crop Type Requirements Prototype	34
Figure 4: Get Farm Status Activity Diagram	35
Figure 5: Manage Crop Type Activity Diagram	36
Figure 6: Manage Crop Type Sequence Diagram	37
Figure 7: Get Farm Status Sequence Diagram	38
Figure 8: Elixir Analysis Class Diagram	39

1. Requirements

1.1. Functional Requirements

1.1.1. Use Case Diagram



Toggle

Figure 1: Elixir Use Case Diagram

1.1.2. Use Case Narratives

Use Case Name:	Login	
Use Case ID:	1	
Priority:	High	
Primary Actor:	User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a user logging into the system. The user's credentials will be validated and upon validation, the user will be authenticated to the system.	
Precondition:	The web service is online.	
Trigger:	This use case will be initiated when a user selects the option to login to the system.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The user requests the option to login.</p> <p>Step 3: The user provides his/her email address and password.</p>	<p>Step 2: The system responds by prompting the user to enter an email address/password combination.</p> <p>Step 4: The system checks for the email address/password combination.</p> <p>Step 5: The system authenticates the user to the system.</p>
Alternate Courses:	Alt-Step 4: If the provided email address and password combination does not exist in the system, notify the user. (go to step 2)	
Conclusion:	This use case concludes when the user is authenticated to the system.	
Postcondition(s):	The user is authenticated to the system.	

Use Case Name:	Change Account Details	
Use Case ID:	2	
Priority:	Medium	
Primary Actor:	User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a user changing his/her account details, specifically his/her email address or password.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a user selects the option to change his/her account details.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user requests the option to change his/her account details	Step 2: The system responds by prompting the user to change either his/her password or email address.
	Step 3: The user requests the option to change his/her email address. Step 5: The use requests the option to change his/her password	Step 4: Extension use case scenario "Change email Address" starts Step 6: Extension use case scenario "Change password" starts
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the user receives confirmation for the changes.	
Postcondition(s):	The user's account details are changed.	

Use Case Name:	Change email Address	
Use Case ID:	3	
Priority:	Medium	
Primary Actor:	User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a user changing his/her email address.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a user selects the option to change his/her email address.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The user requests the option to change his/her email address.</p> <p>Step 3: The user provides a new email address and his/her current password.</p>	<p>Step 2: The system responds by showing the user his/her current email address and prompting for a new email address along with his password.</p> <p>Step 4: The system checks if the provided password is valid for the current user.</p> <p>Step 5: The system changes current user's email address with the user provided one.</p> <p>Step 6: The system displays a confirmation for the e-mail address change.</p>
Alternate Courses:	Alt-Step 4: If the provided password is not valid for the current user, notify the user (go to step 2).	
Conclusion:	This use case concludes when the user receives confirmation for the e-mail address change.	
Postcondition(s):	The user's email address is changed.	

Use Case Name:	Change Password	
Use Case ID:	4	
Priority:	Medium	
Primary Actor:	User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a user changing his/her password.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a user selects the option to change his/her password.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The user requests the option to change his/her password.</p> <p>Step 3: The user provides a new password and his/her current password.</p>	<p>Step 2: The system responds by prompting the user for a new password along with his/her current password.</p> <p>Step 4: The system checks if the provided password is valid for the current user.</p> <p>Step 5: The system changes current user's password with the user provided one.</p> <p>Step 6: The system displays a confirmation for the password change.</p>
Alternate Courses:	Alt-Step 4: If the provided password is not valid for the current user, notify the user (go to step 2).	
Conclusion:	This use case concludes when the user receives confirmation for the password change.	
Postcondition(s):	The user's password is changed.	

Use Case Name:	Manage Regular User	
Use Case ID:	5	
Priority:	Medium	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user managing a regular user by creating, editing and deleting.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to manage a regular user.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin user requests the option to manage a regular user. Step 3: The admin user requests one of the options.	Step 2: The system responds by providing the admin user the option to create (UC #6), edit (UC #7) or delete (UC #8) regular users. Step 4: Selected option's use case starts.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the admin user receives confirmation for the user information change.	
Postcondition(s):	The user's information is changed.	

Use Case Name:	Create Regular User	
Use Case ID:	6	
Priority:	Medium	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user creating a regular user with adding name and email.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to create a regular user.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The admin user requests the option to create a regular user.</p> <p>Step 3: The admin user provides a name and email address.</p>	<p>Step 2: The system responds by prompting the admin user for a name and email address</p> <p>Step 4: The system checks that the provided email address is not in use.</p> <p>Step 5: The system creates a new regular user along with a randomly generated password.</p> <p>Step 6: The system displays a confirmation for the created user.</p>
Alternate Courses:	Alt-Step 4: If the provided email address already exists, notify the admin user (go to step 2).	
Conclusion:	This use case concludes when the admin user receives confirmation for the created user.	
Postcondition(s):	A new regular user is created.	

Use Case Name:	Edit Regular User	
Use Case ID:	7	
Priority:	Medium	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user editing a regular user information including email, password and name.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to edit a regular user.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The admin user requests the option to edit a regular user.</p> <p>Step 3: The admin user selects a user.</p> <p>Step 6: The admin user provides the new user information.</p>	<p>Step 2: The system responds by listing all the registered users in the system, prompting the admin user to select a user.</p> <p>Step 4: The system responds by printing the user's name and email address.</p> <p>Step 5: The system prompts the admin user for optional fields of name and email address.</p> <p>Step 7: The system checks that the provided email address is not in use.</p> <p>Step 8: The system updates the selected user's information.</p> <p>Step 9: The system displays a confirmation for the updated user.</p>

Alternate Courses:	Alt-Step 7: If the provided email address already exists, notify the admin user (go to step 4).
Conclusion:	This use case concludes when the admin user receives confirmation for the edited user information.
Postcondition(s):	The selected regular user's information is updated.

Use Case Name:	Delete Regular User	
Use Case ID:	8	
Priority:	Medium	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user deleting a regular user.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to create a regular user.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The admin user requests the option to delete a regular user.</p> <p>Step 3: The admin user selects a user.</p> <p>Step 5: The admin user confirms the deletion of the user.</p>	<p>Step 2: The system responds by listing all the registered users in the system, prompting the admin user to select a user.</p> <p>Step 4: The system prompts the admin user for the confirmation of the deletion.</p> <p>Step 6: The system deletes the selected user.</p> <p>Step 7: The system displays a confirmation for the deleted user.</p>
Alternate Courses:	Alt-Step 5: If the admin user doesn't confirm the deletion, notify the admin user (go to step 2).	
Conclusion:	This use case concludes when the admin user receives confirmation for the deleted user.	
Postcondition(s):	The selected user is deleted.	

Use Case Name:	Manage Farm	
Use Case ID:	9	
Priority:	High	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user managing a farm by creating, editing and deleting a farm.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to manage a farm.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin user requests the option to manage a farm. Step 3: The admin user requests one of the options.	Step 2: The system responds by providing the admin user the option to create (UC #10), edit (UC #11) or delete (UC #12) a farm. Step 4: Selected option's use case starts.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the admin user receives confirmation for the farm information change.	
Postcondition(s):	The farm's information is changed.	

Use Case Name:	Create Farm	
Use Case ID:	10	
Priority:	High	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user creating a farm by editing the farm name, location, crop type and the associated user.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to create a farm.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The admin user requests the option to create a farm.</p> <p>Step 3: The admin user provides the farm name, location, crop type and the associated user.</p>	<p>Step 2: The system responds by prompting the admin user for the farm name, location, crop type and the associated user.</p> <p>Step 4: The system checks that the provided email address is not in use.</p> <p>Step 5: The system creates the farm and associates the provided user with it.</p> <p>Step 6: The system displays a confirmation for the created farm.</p>
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the admin user receives confirmation for the created farm.	
Postcondition(s):	A new farm is created.	

Use Case Name:	Edit Farm	
Use Case ID:	11	
Priority:	High	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user editing a farm by changing the farm name, location, crop type and the associated user.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to edit a farm.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin user requests the option to edit a farm.	Step 2: The system responds by listing all the farms in the system, prompting the admin user to select a farm.
	Step 3: The admin user selects a farm.	Step 4: The system responds by printing the farm name, location, crop type and the associated user.
	Step 5: The admin user provides the new farm information.	Step 6: The system prompts the admin user for optional fields of farm name, location, crop type and the associated user.
		Step 7: The system updates the selected farm's information. Step 8: The system displays a confirmation for the edited farm.
Alternate Courses:	N/A	

Conclusion:	This use case concludes when the admin user receives a confirmation for the edited farm information.
Postcondition(s):	The selected farm's information is updated.

Use Case Name:	Delete Farm	
Use Case ID:	12	
Priority:	High	
Primary Actor:	Admin User	
Other Actor(s):	N/A	
Description:	This use case describes the event of an admin user deleting a farm.	
Precondition:	The web service is online. The user is logged into the system. Farms must be created first for deleting operations.	
Trigger:	This use case will be initiated when an admin user selects the option to delete a farm.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin user requests the option to delete a farm.	Step 2: The system responds by listing all the farms in the system, prompting the admin user to select a user.
	Step 3: The admin user selects a farm.	Step 4: The system prompts the admin user for the confirmation of the deletion.
	Step 5: The admin user confirms the deletion of the farm.	Step 6: The system deletes the selected farm. Step 7: The system displays a confirmation for the deleted farm.
Alternate Courses:	Alt-Step 5: If the admin user doesn't confirm the deletion, notify the admin user (go to step 2).	
Conclusion:	This use case concludes when the admin user receives a confirmation for the deleted farm.	
Postcondition(s):	The selected farm is deleted.	

Use Case Name:	Manage Manual Irrigation	
Use Case ID:	13	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the events of a regular managing manual irrigation settings.	
Precondition:	The web service is online. The regular user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to manage his/her farm's manual irrigation options.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to manage manual irrigation. Step 3: The regular user requests one of the options.	Step 2: The system responds by providing the regular user the option to start (UC #14), stop(UC #15) or schedule (UC #16) irrigation. Step 4: Selected option's use case starts.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives confirmation for the irrigation settings.	
Postcondition(s):	The user's manual irrigation settings are changed.	

Use Case Name:	Start Irrigation	
Use Case ID:	14	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a regular user starting irrigation.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when an admin user selects the option to start irrigation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to start irrigation.	Step 2: The system starts the irrigation. Step 3: The system displays a confirmation for the started irrigation to the user.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives a confirmation for the started irrigation.	
Postcondition(s):	The regular user's farm start to irrigate.	

Use Case Name:	Stop Irrigation	
Use Case ID:	15	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a regular user stopping irrigation.	
Precondition:	The web service is online. The user is logged into the system. The start irrigation option must be triggered.	
Trigger:	This use case will be initiated when an admin user selects the option to start irrigation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to stop irrigation.	Step 2: The system stops the irrigation. Step 3: The system displays a confirmation for the stopped irrigation to the user.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives a confirmation for the stopped irrigation.	
Postcondition(s):	The regular user's farm stops irrigation.	

Use Case Name:	Schedule Irrigation	
Use Case ID:	16	
Priority:	High	
Primary Actor:	Regular user	
Other Actor(s):	N/A	
Description:	This use case describes the event of a regular user scheduling an irrigation.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to schedule irrigation.	
Typical Course of Events:	Actor Action	System Response
	<p>Step 1: The regular user requests the option to schedule irrigation.</p> <p>Step 3: The regular user selects the time and days for the scheduled irrigation.</p>	<p>Step 2: The system responds by prompting the admin user for irrigation days and time frame.</p> <p>Step 4: The system checks that the provided schedule is in a valid time frame.</p> <p>Step 5: The system creates an irrigation schedule.</p> <p>Step 6: The system displays a confirmation for the scheduled irrigation to the user.</p>
Alternate Courses:	Alt-Step 4: If the provided schedule is not in a valid time frame, notify the regular user.. (go to step 2).	
Conclusion:	This use case concludes when the regular user receives a confirmation for the scheduled irrigation.	
Postcondition(s):	The regular user creates the schedule for irrigation.	

Use Case Name:	Start Scheduled Irrigation	
Use Case ID:	17	
Priority:	High	
Primary Actor:	Timer	
Other Actor(s):	N/A	
Description:	This use case describes the event of a timer starting a scheduled irrigation.	
Precondition:	The web service is online. There is scheduled irrigation for a farm.	
Trigger:	This use case will be initiated when the system time is within a scheduled irrigation time frame.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The timer requests the option to start the irrigation for the user-specified length when the system time reaches the scheduled irrigation time.	Step 2: The system starts the irrigation.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the system starts the scheduled irrigation.	
Postcondition(s):	The scheduled irrigation starts for the user-specified time frame.	

Use Case Name:	Manage Auto Irrigation	
Use Case ID:	18	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the events of a regular managing auto irrigation settings.	
Precondition:	The web service is online. The regular user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to manage his/her farm's automatic irrigation options.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to manage automatic irrigation. Step 3: The regular user requests one of the options.	Step 2: The system responds by providing the regular user the option to start (UC #19) and stop (UC #20) auto irrigation. Step 4: Selected option's use case starts.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives confirmation for the irrigation settings	
Postcondition(s):	The user's automatic irrigation settings are changed.	

Use Case Name:	Start Auto Irrigation	
Use Case ID:	19	
Priority:	Medium	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a regular user turning on the auto irrigation setting.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to start auto irrigation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to start auto irrigation	Step 2: The system turns on the auto irrigation setting. Step 3: The system displays a confirmation for the irrigation setting to the user.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives a confirmation for the irrigation setting.	
Postcondition(s):	The auto irrigation system is turned on.	

Use Case Name:	Stop Auto Irrigation	
Use Case ID:	20	
Priority:	Medium	
Primary Actor:	Regular User	
Other Actor(s):	N/A	
Description:	This use case describes the event of a regular user turning off the auto irrigation setting.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to stop auto irrigation.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to stop auto irrigation	Step 2: The system turns off the auto irrigation setting. Step 3: The system displays a confirmation for the irrigation setting to the user.
Alternate Courses:	N/A	
Conclusion:	This use case concludes when the regular user receives a confirmation for the irrigation setting.	
Postcondition(s):	The auto irrigation system is turned off.	

Use Case Name:	Auto Irrigation	
Use Case ID:	21	
Priority:	Medium	
Primary Actor:	Sensors	
Other Actor(s):	Agricultural Weather Forecast, Meteorology API	
Description:	This use case describes the event of the automated irrigation.	
Precondition:	The web service is online. Auto irrigation is turned on.	
Trigger:	This use case will be triggered when the irrigation conditions are met for the farm.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The sensors request the option to start the irrigation when they meet the optimal levels.	Step 2: The system queries the agricultural weather forecast system and the meteorology API for the temperature, heat stress and the frost risk. Step 3: The system verifies that the meteorological conditions are met for irrigation Step 4: The system starts the irrigation.
Alternate Courses:	Alt-Step 3: If the meteorological conditions are not met for the system, do not start the irrigation, terminate the use case.	
Conclusion:	This use case concludes when the irrigation starts.	
Postcondition(s):	The regular user's farm starts to irrigate.	

Use Case Name:	Get Farm Status	
Use Case ID:	22	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	Meteorology API, Sensors, Agricultural Weather Forecast	
Description:	This use case describes the event of a regular user getting his/her farm information.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be triggered when a regular user presses the get farm status option.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to get farm status.	Step 2: The system queries the sensors' for the light, temperature and the soil moisture levels. Step 3: The system queries the agricultural weather system for the heat stress, frost risk, pesticide eligibility. Step 4: The system queries the meteorology API for the temperature and the precipitation levels. Step 5: The system parses and visualizes the output.
Alternate Courses:	Alt-Step 2: If one or more of the sensors cannot be reached, display an error message. Alt-Step 3: If the agricultural weather system cannot be reached, display an error message. Alt-Step 4: If the meteorology API cannot be reached, display an error message.	

Conclusion:	This use case concludes when the farm status data is visualized.
Postcondition(s):	The farm status data is acquired, parsed and visualized.

Use Case Name:	Get Crop Info	
Use Case ID:	23	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	Meteorology API, Sensors	
Description:	This use case describes the event of a regular user getting his/her crop information.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be initiated when a regular user selects the option to get crop information.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to get crop information.	Step 2: The system acquires the sensor values and the meteorology data via the meteorology API for the user's farm location. Step 3: The system parses the acquired data and compares the output with the optimal levels Step 4: The system visualizes the output.
Alternate Courses:	Alt-Step 2: If one or more of the sensors or the meteorology API cannot be reached, display an error message. Terminate the use case.	
Conclusion:	This use case concludes when the crop status data is visualized.	
Postcondition(s):	The crop info is gathered and visualized.	

Use Case Name:	Manage Crop Type	
Use Case ID:	24	
Priority:	High	
Primary Actor:	Regular User	
Other Actor(s):	Agricultural Weather Forecast	
Description:	This use case describes the event of a regular user changing his/her crop type.	
Precondition:	The web service is online. The user is logged into the system.	
Trigger:	This use case will be triggered when a regular user selects the option to manage crop type.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The regular user requests the option to manage crop type.	Step 2: The system responds by showing the available crop types and prompts the user to select a crop type.
	Step 3: The regular user selects a crop type.	Step 4: The system responds by showing the user information regarding the selected crop type
	Step 6: Regular user confirms the new crop type.	Step 5: The system prompts the regular user for the confirmation of the crop type change. Step 7: The system changes the crop type. Step 8: The system displays a confirmation for the crop type change.
Alternate Courses:	Alt-Step 6: If the regular doesn't confirm the change, notify the user. Terminate the use case.	
Conclusion:	This use case concludes when the regular user receives confirmation for the crop type change.	

Postcondition(s):	The regular user changes the crop type for his/her farm.
--------------------------	--

1.2. Non-Functional Requirements:

1.2.1. Performance

- The system shall display the visualized output of farm and crop information to the user in less than 3 seconds.

1.2.2. Security

- The system shall communicate over encrypted channels for both the web application and the MQTT devices.
- All the sensitive data shall be encrypted both in rest and in transit.

1.2.3. Reliability

- The devices within the MQTT network shall have the fault tolerance mechanisms to restart both the devices and the services when they stop working.
- Web servers should be online and can be reachable.

2. Constraints:

- The web application will be a progressive web application intended to work on both desktop and mobile devices.
- The Node.js runtime environment along with the React framework will be used for the web application.
- The NoSQL database MongoDB will be used for the user data and the web application and the time-series database InfluxDB will be used for the sensor data.
- All updates to the codebase will be done using Git in order not to lose data and manage versions more efficiently.
- In order to maintain a persisting code style, standards such as *Standard* will be followed and team members will be following a predetermined coding style.
- A list of security practices will be determined and followed by the developers of the project to prevent API key and source code leaks.

3. Requirements Prototype:

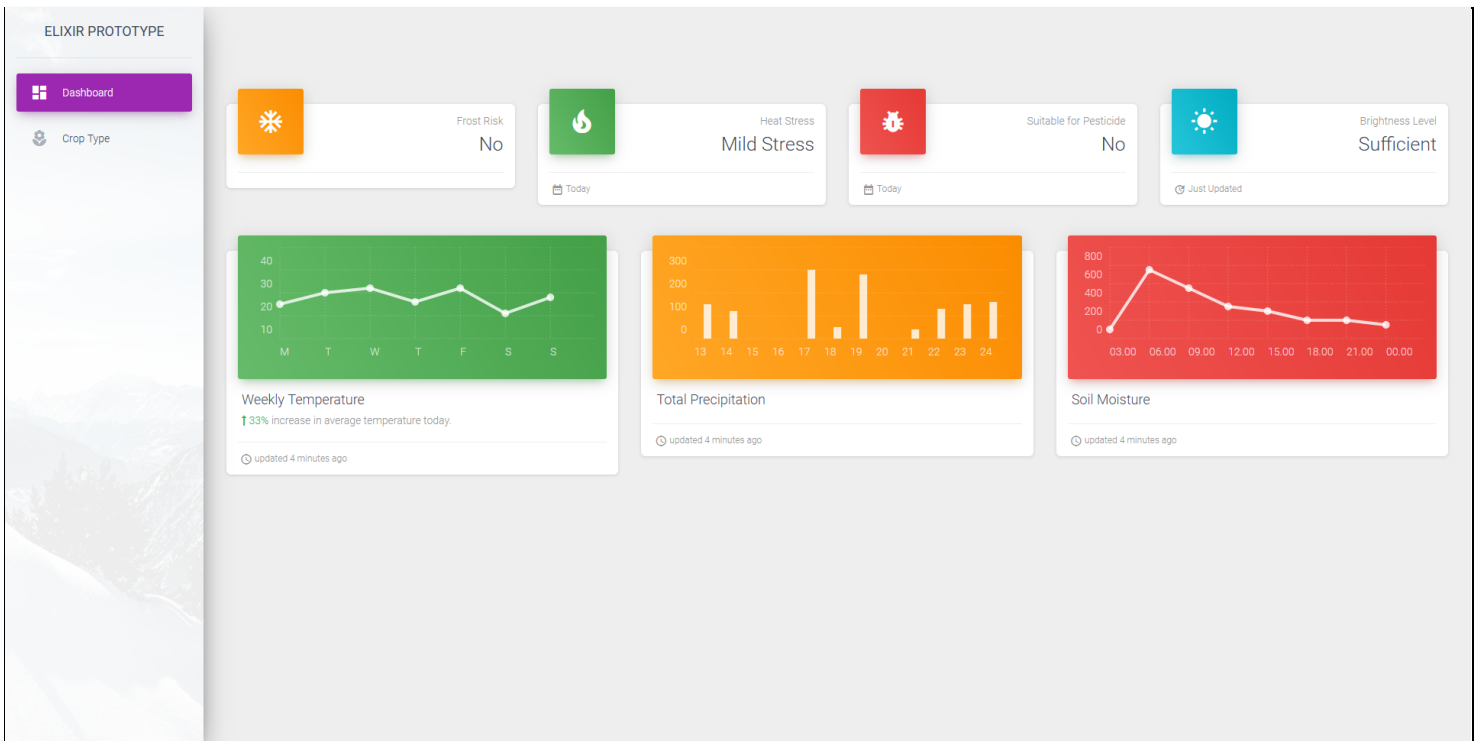


Figure 2: Get Farm Status Requirements Prototype

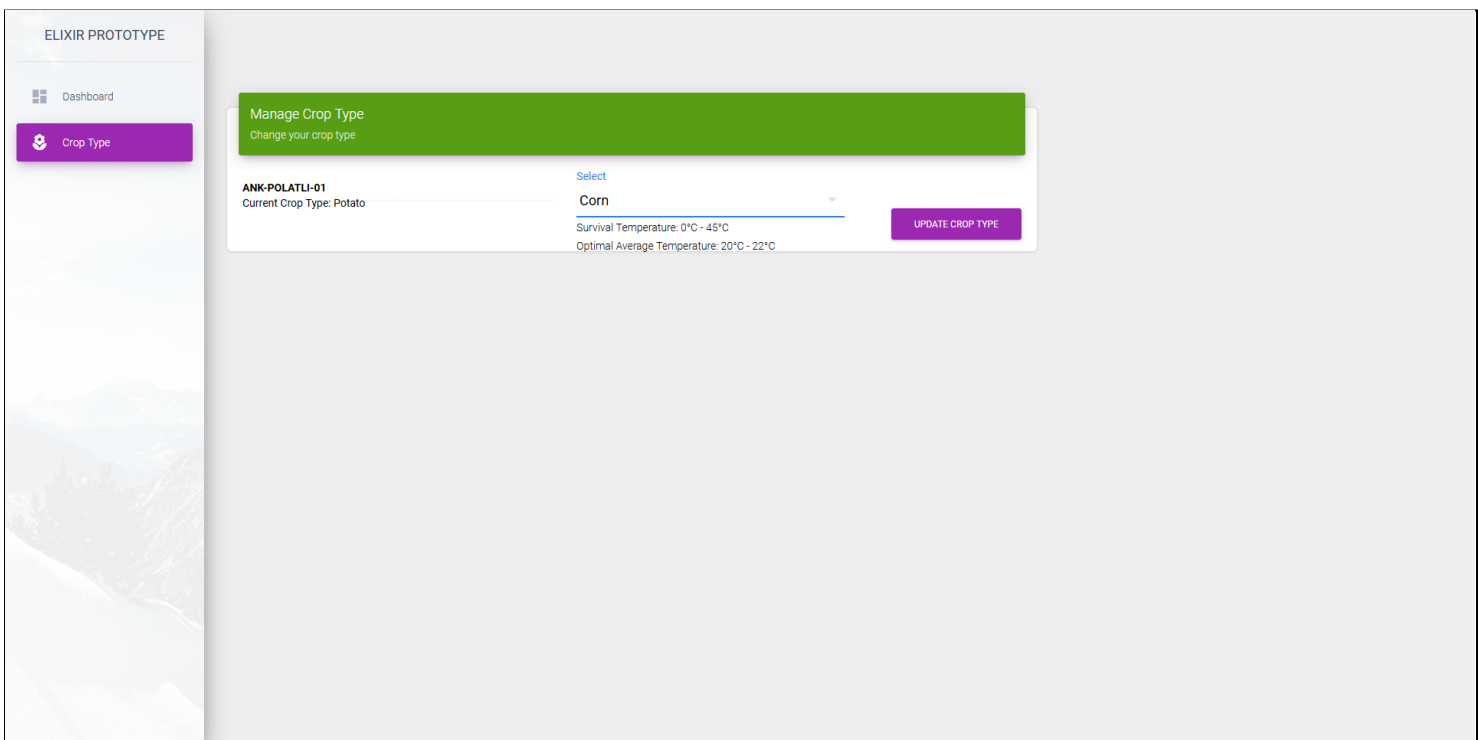


Figure 3: Manage Crop Type Requirements Prototype

4. System Models:

4.1. Activity Diagrams:

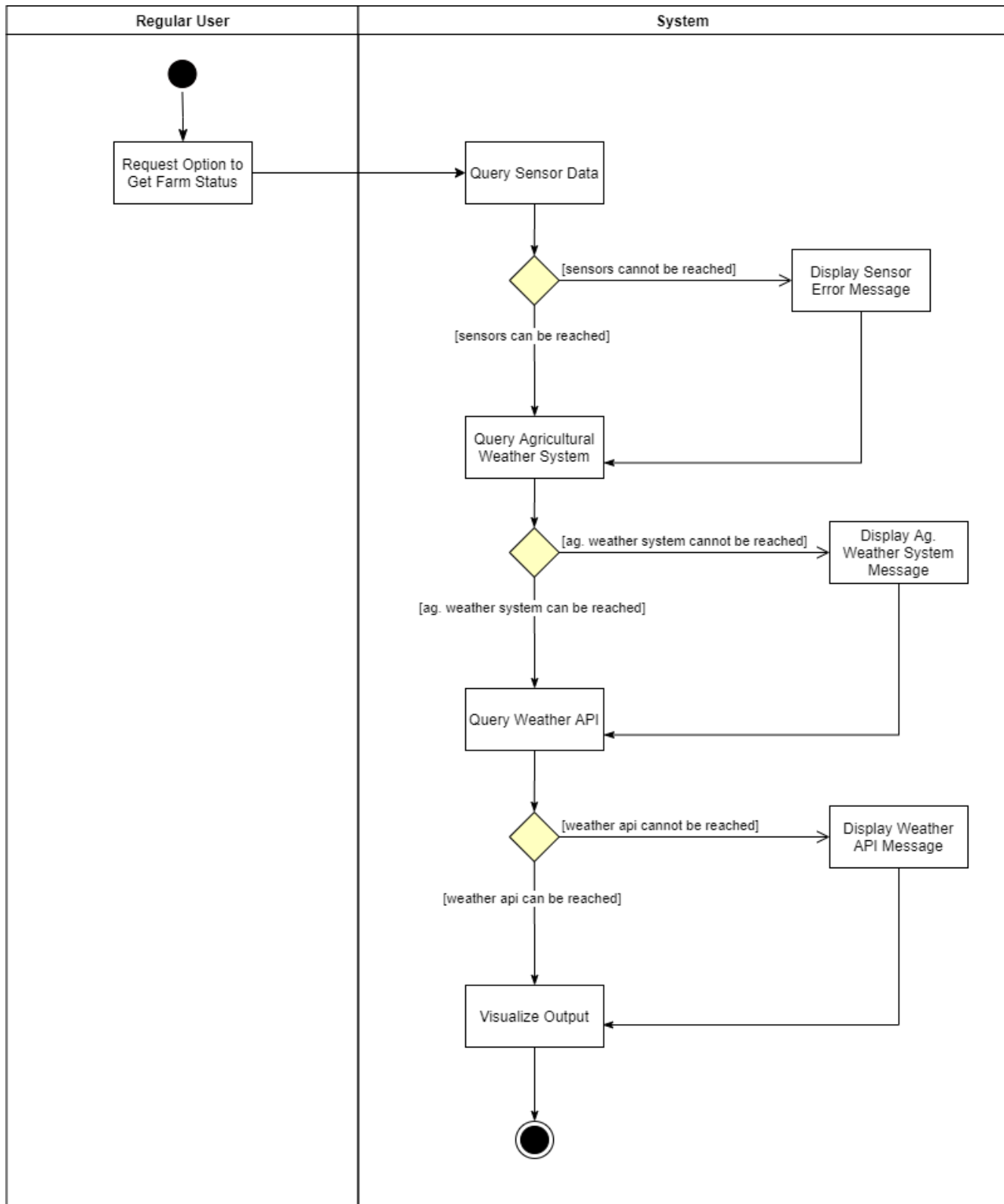


Figure 4: Get Farm Status Activity Diagram

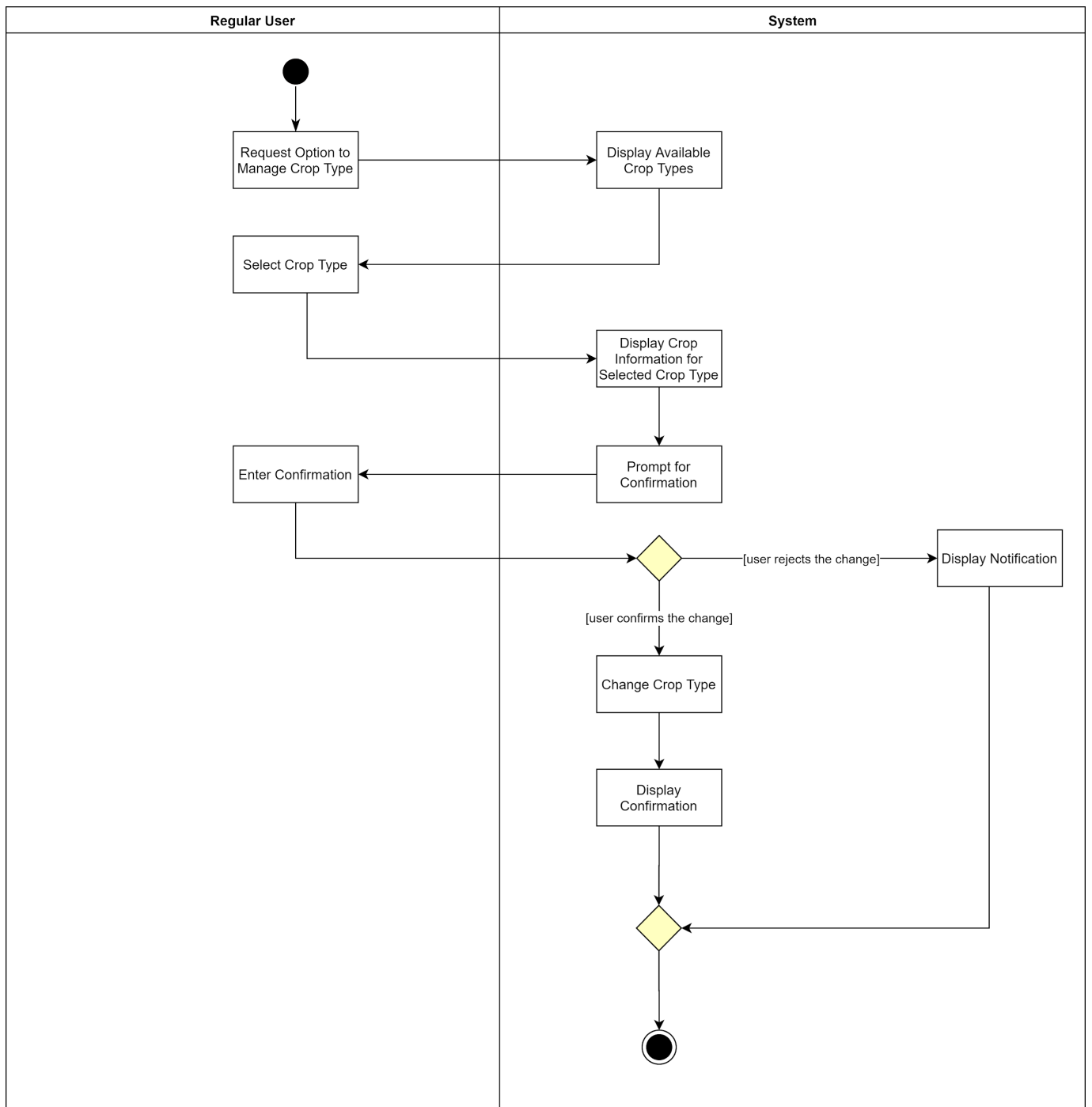


Figure 5: Manage Crop Type Activity Diagram

4.2. System Sequence Diagrams:

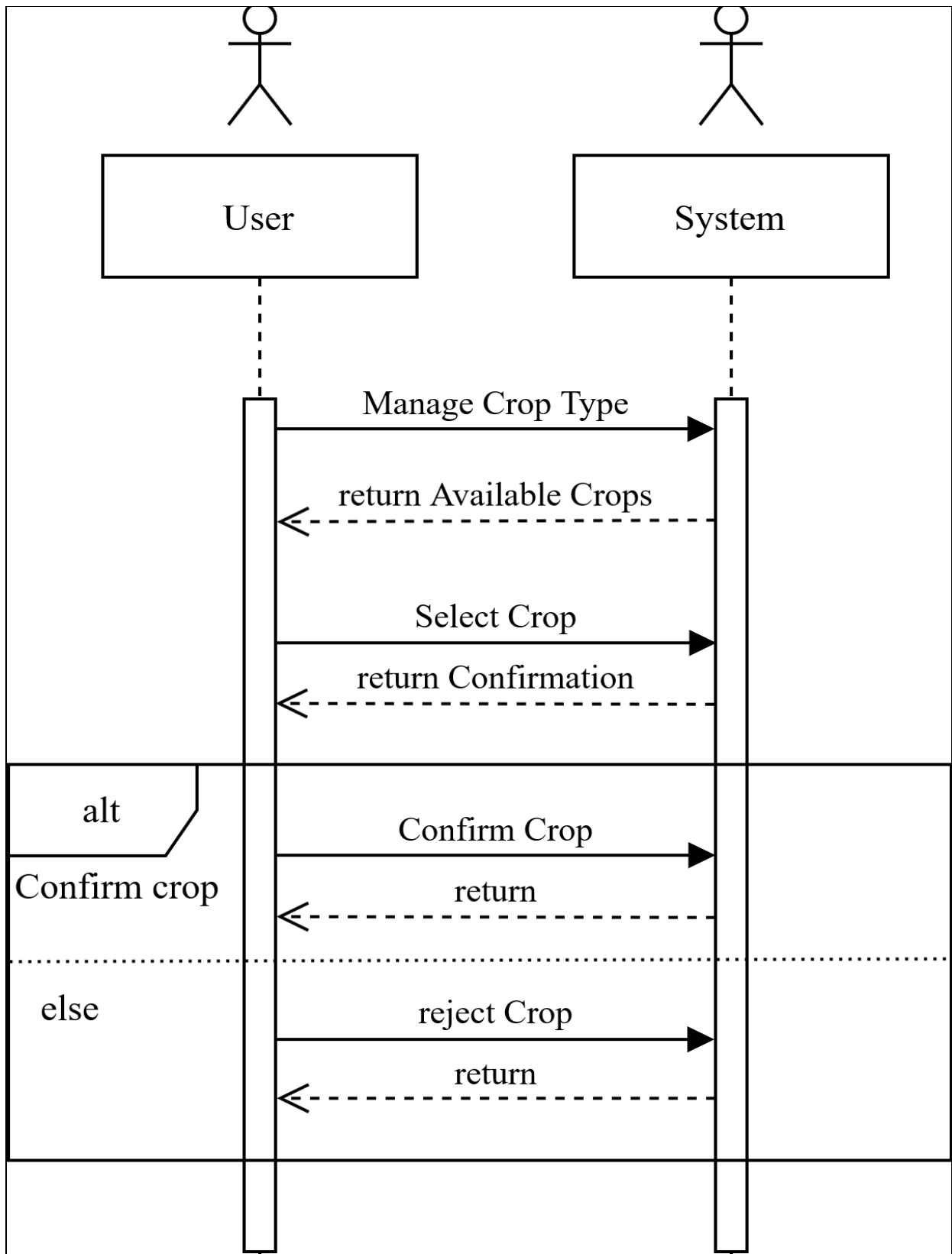


Figure 6: Manage Crop Type Sequence Diagram

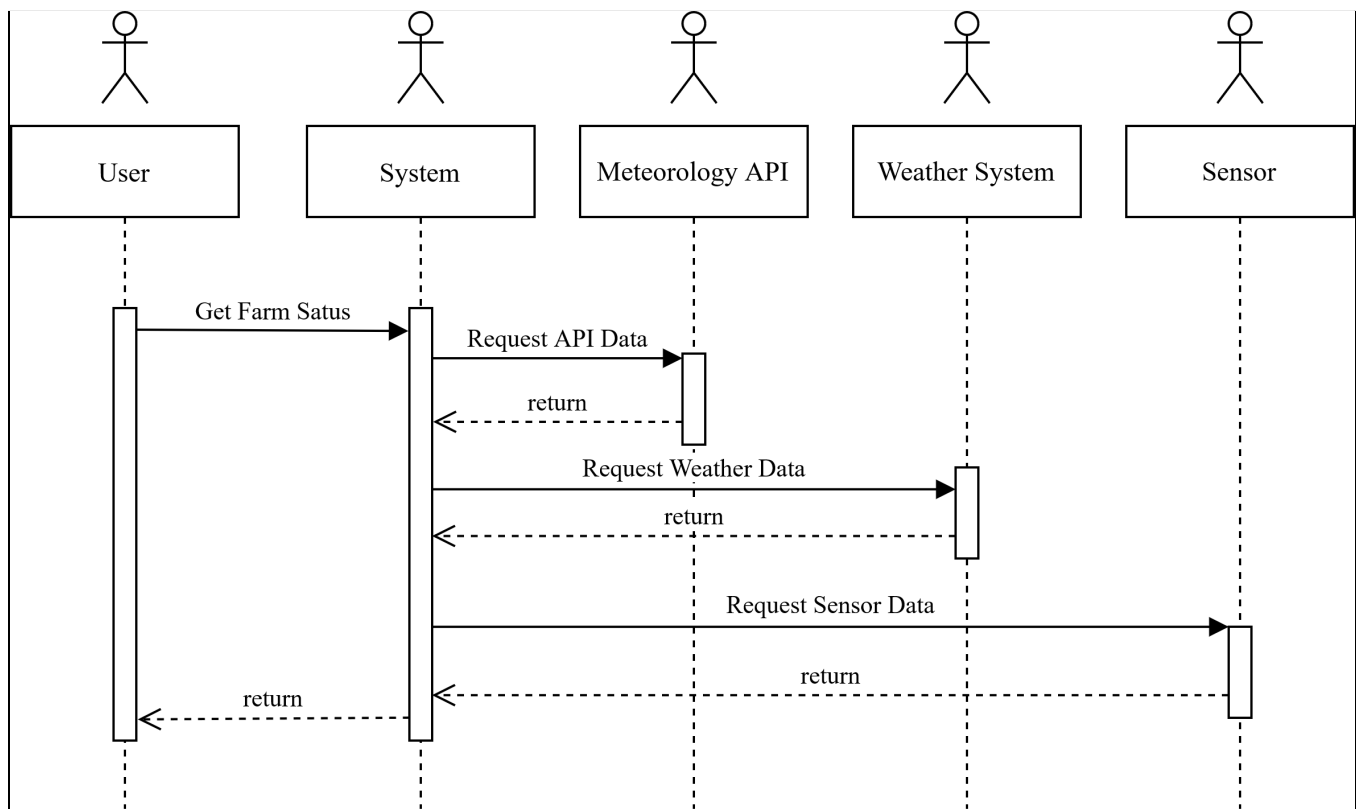


Figure 7: Get Farm Status Sequence Diagram

4.3. Analysis Class Diagram:

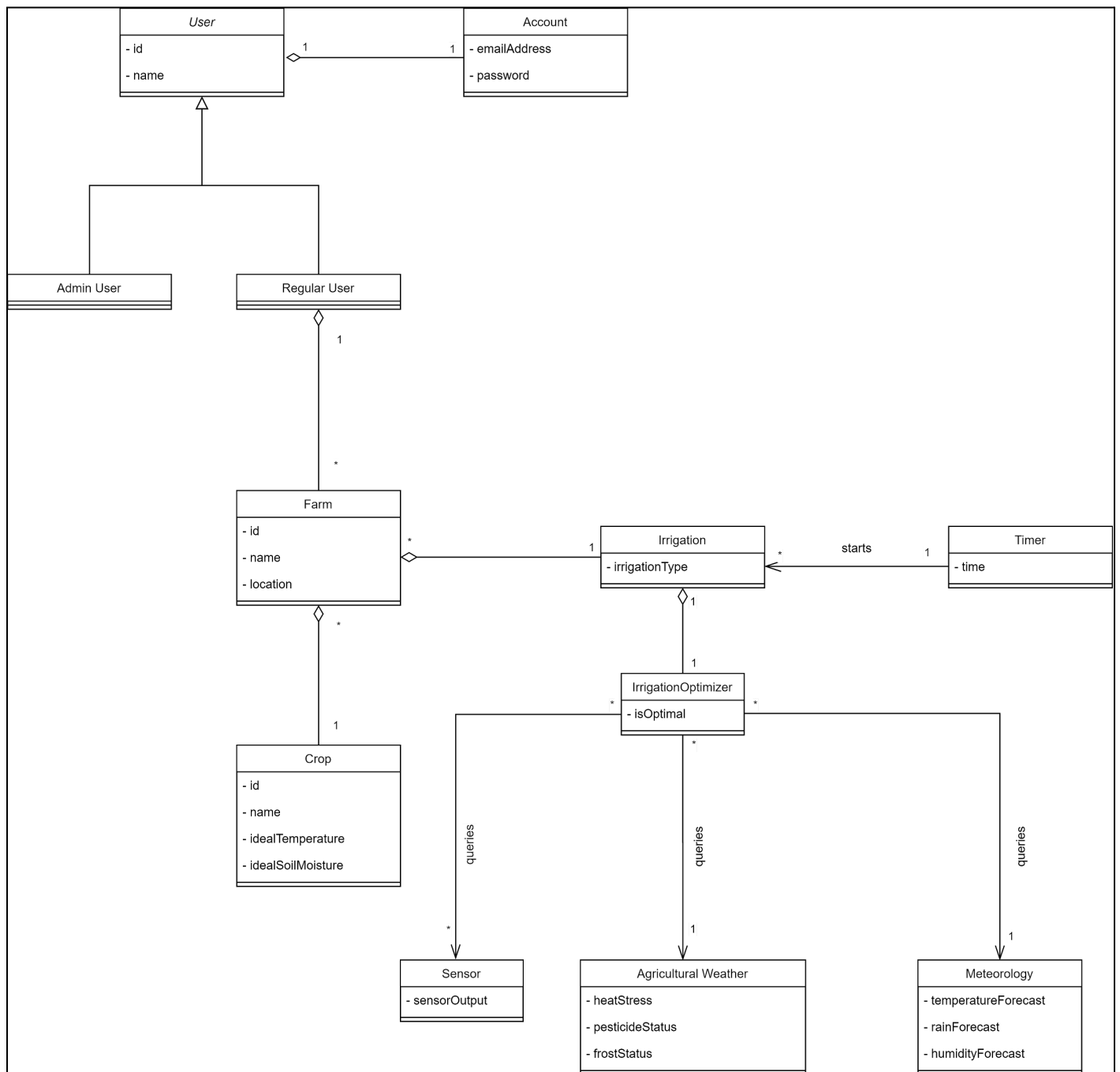


Figure 8: Elixir Analysis Class Diagram