Farm Management System Requirements Specification

Version 1.3

June 14, 2021

To regenerate the TOC, select all (CTL-A) and press F9.

Table of Contents

1. Executive Summary

1.1 Project Overview

In the last 10 years farms in Albania had a noticeable development after years of forced collectivization of agricultural cooperatives. According to official statistics, today in Albania, more than 90% of farms are mainly based on family farms that have an economic and social dimension; that is around 300 000 farms. Our goal is to provide the local farmers with an efficient solution based on their needs. Our project is centered on providing a management system for farmers in order to ease the management of their daily activity.

The Farming Management System will help to plan, monitor and analyze all activities on the farm easily. Tillage, planting, crop protection, fertilization, irrigation, harvesting and all other activities are managed with a few clicks. Plus, you can track input usage quantities, costs and work hours for every activity. The system will be also centered around livestock management: that include buying livestock, selling, and observing or calculating profit. Whether you are a cattleman or an agriculturist our product will provide useful facilities for you to control your business.

1.2 Purpose and Scope of this Specification

The purpose of this project is to implement technology in order to improve and facilitate the everyday operations of farms. This document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external actions.

This document is intended for both the stakeholders and the developers of the system.

This software system will be a Farming Management System for local farm owners. This system will be designed to support production management and to comply with actual farming standards. By maximizing the farmer's work efficiency and production, the system will meet the farmer's needs while remaining easy to understand and use.

2. Product/Service Description

According to INSTAT, in Albania, as of 2011 there were 353'341 farms. Out of these farms, 299'077 were farms that practiced both livestock farming and agriculture, 54'086 were farms that practiced only agriculture, and 178 practiced only livestock breeding and farming.

2.1 Product Context

This product is independent and self-contained.

2.2 User Characteristics

Users that will use the program are:

- *Farmers/Owners*: the main users for which this program is built, they are often inexperienced with use of such software so we must make it easy for them.

- *Farm economists*: these users will use the system heavily in calculating the profit and the spending of a farm. They expertise in the economic aspect of the product and to them will come in handy graphs that we will provide.

- *Employees*: are the workforce of the farm, they will have a simple UI where mostly they will be assigned a task and they will report on it.

- *Farm managers*: in big farms the role of the owner and the role of farm manager may be done by different people. Farm managers are concerned with the economic aspect as well as with the organizational aspect of a farm.

2.3 Assumptions

Assumptions that affect the requirements are:

- The system must be available at all times.

- The user must be able to change and manipulate the data they enter in real time.

- There will be multiple users.

- There are 3 different scenarios in which the system may be used: a farm that practices both livestock farming and agriculture, a farm that only practices agriculture and a farm that practices only livestock breeding and farming.

2.4 Constraints

User constrains:

- One user will only be able to see its own data and no other users' data.

Framework constrains:

- Laravel is easy to learn and fast to develop but some features are clearly better-suited for rapid prototyping and should be replaced as an application grows.

- Symfony on the other hand seems to be a better option to consider in the future.

3. Requirements

3.1 Functional Requirements

BR => Business Requirement

BR_A => Agriculture only farm

BR_L => Livestock only farm

BR_AL => Agriculture and Livestock

Note: BR_A and BR_L apply to BR_AL as well. BR applies to all of them.

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_01	The system should store and show the data in an organized and consistent way.	Users must have access to all their data all of the time	1	06-Jun-21	Kejdi Domi
BR_02	The system must provide notifications between users.	Users can send messages to each other.	1	06-Jun-21	Kejdi Domi
BR_03	The system must prompt the user for registering its data when they are firstly using the system; registering a new account.	This should be handled with care and must be a multistep process so the user will not get bored and tired.	2	06-Jun-21	Kejdi Domi
BR_04	The only user which can be registered to the system autonomously will be the manager.	He will add other users as the registering phase continues.	1	06-Jun-21	Kejdi Domi
BR_05	After the manager creates a new user which may be an Owner, an Economist, or an Employee, the system must inform that user by email, including a temporary password in the email.	Logical conclusion of BR_04.	1	06-Jun-21	Kejdi Domi

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_AL_06	The system must keep a list of Livectock information and Crop information that the farm sells/manages.	This will be shown in the dashboard like Crops, Livestock.	1	06-Jun-21	Kejdi Domi
BR_A_07	The system must provide curated information about the crops divided in these subsections: When to plant, How to take care, and Economic aspect	Research to be done for the types of plants and agricultural aspect. Information accessible in learn more section of Crops view in dashboard.	2	06-Jun-21	Kejdi Domi
BR_L_08	The system must provide an interface to manage veterinarians, where a user can view information like availability, address, phone number, name, and email.	We must account for multiple veterinarians.	2	06-Jun-21	Kejdi Domi
BR_L_09	The system must notify the user for upcoming vaccines or medical treatments that Livestock must go through their lifetime.	This is important to the certification of the institution and must be handled professionally.	1	06-Jun-21	Kejdi Domi
BR_AL_10	The system must keep a list of all buyers and suppliers of Livestock and Crops.	They will collectively be called The Market, and will be included in dashboard.	2	06-Jun-21	Kejdi Domi
BR_AL_11	The system must show best buyers; this will be calculated as a weighted average of the amount they buy, the cost they are willing to buy and the frequency.	This information together with BR_AL_10 is important to users: Manager, Economist and Owner.	3	06-Jun-21	Kejdi Domi

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_AL_12	The system must show the most profitable suppliers; this will be calculated as a weighted average of the price they are willing to sell, the distance from the farm and the prosperity of the calves bought.	We will prompt the manager to add the distance in km.	3	06-Jun-21	Kejdi Domi
BR_13	The system must provide a comprehensive and detailed description of the legal requirements a farm must have to function properly in Albania.	We must learn and get used to all the legislation concerning farms in our country. This will be included in Legal Information in the dashboard.	1	06-Jun-21	Kejdi Domi
BR_14	At the end of every month the Economist must complete a Monthly Report form detailing the transactions and concluding in profit calculation.	This requirement is purely economic and is relied entirely on Economist actor.	1	06-Jun-21	Kejdi Domi
BR_A_15	Show a tracking page for plants that the user planted in form of a loading bar which will show the user the right time for seed planting/transplant and the right time for harvest.	The loading bar should have a gradient to it from green to red. The page will be accessed through Track button at Crop interface in the dashboard.	3	06-Jun-21	Kejdi Domi

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_16	The system must keep a list of all users: Owners, Managers, Economists and Employees.	Certain users have privileges over others, for example: Owner can delete, add, modify Manager and Employee, Manager can delete, add, modify Economist and Employee but can only add Owner (connected to BR_04), Economist has only modify privileges over certain aspects of Employee user like wage. Employees can be considered least privileged users.	1	06-Jun-21	Kejdi Domi
BR_17	The system must provide an interface for the Manager where they can assign tasks to Employees.	This can be implemented as a button in the Employee entry in the Employee table (Employees Dashboard) which redirects to the task form. Keep in mind tasks can be repeated through the day, week, or even months. (Example => Time - 21:00, Repeat – every day, Task – clean the calves. Deniable - no)	2	06-Jun-21	Kejdi Domi
BR_18	The system must notify Employees for new tasks that were assigned to them.	In the Tasks section in employee dashboard.	2	06-Jun-21	Kejdi Domi
BR_19	Employees must be prompted to complete a short report on the tasks they accomplish.	If the task is deniable, they can explain why they denied it.	3	06-Jun-21	Kejdi Domi

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_A_20	Suggestions about other uses for the crops the user is harvesting should be shown on loading screens and on the bottom of the homepage.	For example, the user might be selling grapes our system should be able to identify that the user is selling grapes (entered by the user) and suggest that the user sell also wine that comes from grapes.	3	06-Jun-21	Kejdi Domi
BR_AL_21	The system must provide a Share feature: a way to share information in json format between different farms. This information includes Market information, Crops information, Vets information and Livestock information.	The idea is that different farms can communicate with each other, especially exchanging Buyers and Suppliers information. Our system will work with JSON format to create files that will be saved locally on the user’s machine and shared (Note that the system itself doesn’t need a way to share information, the sharing is done by the user and may be done by email.)	3	06-Jun-21	Kejdi Domi
BR_22	The system must handle Deletions of data with caution; for highly important information like deleting a manager the system should ask for the current user’s password.	This is done in order to not lose critical information.	1	06-Jun-21	Kejdi Domi
BR_A_21	Create a database separate from the whole project where we will store all data necessary for the growth and maintenance of different plants	This requirement is very important and needs further specifications. It will be divided into other requirements that will need their own research.	1	19-Apr-21	Kejdi Domi

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_AL_22	Create a tab where the user can create, manage, and view the space it needs to store the crops harvested. The user can add warehouses, haystacks, and barrels.	For example, the user will add a haystack to his/her farm, we will prompt him to enter a value signaling the capacity of the haystack and after this another value that will tell how much of the haystack is occupied/free in percentage.	2	19-Apr-21	Kejdi Domi
BR_AL_23	On startup and on loading pages, suggest crops (like soy) that livestock eats or show curiosities around the world related to livestock farms which can improve the performance of the farm.	A little touch to detail.	3	19-Apr-21	Kejdi Domi
BR_AL_24	Alert the user for the best time to invest in new livestock which will be correlated with the time that the crops are ready for harvest, and the time when there are more weddings.	If you are buying bred cows, the period from about May 15 to August 1 might be a good time, although the volume of sales is typically low.	3	19-Apr-21	Kejdi Domi
BR_25	The system must allow the user to choose its currencies and store the conversion rates between them.	Rates can be entered manually or chosen from dropdown menu.	2	06-Jun-21	Kejdi Domi
BR_26	The system must provide updated converting rates between currencies. Rates are to be updated every 24 hours.	Can be implemented through an API.	2	06-Jun-21	Kejdi Domi
BR_AL_27	The system must provide weather information.	This will be implemented through an API.	2	06-Jun-21	Kejdi Domi
BR_AL_28	The system must update weather information every 24 hours.	Weather will be shown in the Weather section in the dashboard.	2	06-Jun-21	Kejdi Domi

3.2 Non-Functional Requirements

3.2.1 User Interface Requirements

- The system must be able to run full screen on a screen no smaller than 800-600 px.
- The UI components should have readable font sizes of Open Sans font family, no less than 16 px.
- Menus should be divided into logical sections and we are to use Hierarchical Menus.
- Error messages are to be shown as popups above the current screen, except when there are errors connected to form completion, in that case the cursor must move to the entity which caused the error and a simple red highlight should be applied to the text area.

3.2.2 Usability

- The system should be easy to work with and provide minimal efforts to be learnt and explained.
- The users must be prompted with messages displaying and explaining how to use the system. An instruction book must be attached to the system when delivered to the users.
- The system should be intuitive.
- In the case of invalid user operations (e.g. invalid user input in form completion), the system should let the user know about it by displaying an error message.
- CRUD operations should proceed after a second user confirmation.

3.2.3 Performance

3.2.3.1 Capacity

The solution should be capable of serving 1000 concurrent users for 70% of the time and up to 3000 concurrent users during the peak times. The solution should be capable of handling a load of 30,000 queries per minute.

3.2.3.2 Availability

- Maximum permitted number of failures per hour = 2.
- The maintenance should be scheduled, and we must notify the user 1 week prior. Maintenance hours should be scheduled after midnights.

- The system should be available 24/7.
- Each user will have his own authentication and session.
- There will be a backup database and a backup server so that if ever any crash happens there always is a backup and when work is being done the application will never stop being live.

3.2.3.3 Latency

Response Time: For 90% for user interactions the application response times should be <2 seconds. For any transaction having a response time greater than 0.5 sec a suitable progress indicator should be shown to the user. Any user interaction response time should not exceed 10 seconds. In cases of unforeseen delays in responses due to network latency etc., a timeout message should be shown after 15 seconds.

3.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

We are to use Sentry (a listener/handler for errors that asynchronously sends out the error/event to Sentry.io) with Laravel for monitoring errors.

3.2.4.2 Maintenance

If a system crash happens the core application is going to restart. While being in this process we are either going to use the other backup server or show to the page the message that for the moment our page is in maintenance until the moment that the server is up and running again. If there is a problem with the restart and backup does not happen correctly the server will be restarted. Application performance is dependent on:

- Database transaction speed.
- User’s internet bandwidth.
- System specifications like RAM, processor speed, etc. dependent on user’s machine.
- Number of users on the application on the same time.
- The Laravel framework capabilities.

3.2.4.3 Operations

- Every user is expected to manage its own information, log in and of course log out.

- Managers are expected to add quantitative and relative data to the farm (Crops, Livestock, Market, Vets), as well as assign tasks to other employees.

- Economists are expected to manage information concerned with the visualization of profit and expenses.

- Owners are expected to deal with the legal technicalities of the farm and communicate. decisions to managers and economists.

- Employees are expected to perform tasks.

- CRUD functionalities affect all users as well as relationship classes like Crops or Livestock.

3.2.5 System Interface/Integration

The system should be capable of running on most web browser, running technologies such as Mozilla or Chrome.

3.2.5.1 Network and Hardware Interfaces

Connection requests are received from port 443, which uses the HTTPS protocol. For the system to work correctly, each browser should support this connection.

3.2.5.2 Systems Interfaces

There are no other systems interfaces.

3.2.6 Security

3.2.6.1 Protection

Any possible or accidental data entry or removal, modification, or access that could risk data integrity, must be taken in consideration and prevented beforehand. When the user enters its credentials, the following validations are checked before the data is stored in the database:

- Check for a valid *name* and *surname*.

- Check for a valid *NIPT*.

- Check for a valid *phone number*.

- Check for a valid *email*.

- Check for a valid *password*.

Any information stored in the database is considered as sensitive information. Passwords are hashed before they are written in the database using an undecryptable algorithm.

The system uses a client-server architecture where data is centralized, making the system more secure and serving added security to its data.

3.2.6.2 Authorization and Authentication

A function is used to check whether the user has entered the correct credentials. If so, the user grants full access on his/her data, otherwise, access is forbidden.

Operations related to authorization on each level are as follows:

1. Farm owners:
   - A function for managing managers.
   - A function for managing economists.
   - A function for viewing economic reports.
   - A function for viewing farm information.
   - A function for viewing the weather.
   - A function for editing farm information.
2. Farm economist:
   - A function for calculating profit.
   - A function for creating economic reports.
   - A function for managing wages.
3. Farm employee:
   - A function for viewing tasks and marking them as done.
4. Farm manager:
   - A function for managing farm information.

- A function for managing owners, economists, employees, and veterinaries.
- A function for managing crops.
- A function for managing livestock.
- A function for managing the market.
- A function for assigning tasks.

The system makes use of the CSRF token, which protects the system from cross-site request forgery attacks, a type of malicious exploit where unauthorized commands are performed on behalf of an authenticated user.

3.2.7 Data Management

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- data access rules
- data entities and relationships
- integrity constraints
- data retention
- valid range, accuracy, and/or tolerance
- units of measure
- data formats
- default or initial values

3.2.8 Other Non-Functional Requirements

The system cannot take any of the following actions on its own without the user confirmation:

- Register a new user, add an employee's task, or wage.

- Update existing information.

- Remove a user from a system.

3.3 Domain Requirements

- The system covers different features; from storing farm information, managing users, to evaluating profits, etc..

- The system contains sensitive information which should be accessible only by users registered in the system.

- The system does not communicate with any other system.

4. Use Cases and User Scenarios

4.1 Use Cases

Use Case 1

Name	Login
Actor(s)	Owner, Economist, Employee, Manager
Description	User provides his registration ID and password to login.
Precondition	User must be registered in the system.
Postcondition	User enters the system.

Use Case 2

Name	Logout
Actor(s)	Owner, Economist, Employee, Manager
Description	User clicks the Logout button and logs out of the system.
Precondition	User must have been already logged in.
Postcondition	User is redirected to Login page and cannot go back.

Use Case 3

Name	Manage personal information
Actor(s)	Owner, Economist, Employee, Manager
Description	User clicks the Edit Profile button and updates his/her personal information. User clicks the Save button to confirm the changes done.
Precondition	User must be logged in the system.
Postcondition	User is redirected to the Home page.

Use Case 4

Name	Manage managers
Actor(s)	Owner
Description	Owner enters the Manage Users page. Owner can add new managers, edit existing data of a manager, or remove a manager from the system. The database is updated after this operation.
Precondition	Owner must be logged in the system. A list of managers should have been added in the database.
Postcondition	Owner is redirected to the Dashboard page.

Use Case 5

Name	Manage economists
Actor(s)	Owner
Description	Owner enters the Manage Users page. Owner can add new economists, edit existing data of an economist, or remove an economist from the system. The database is updated after this operation.
Precondition	Owner must be logged in the system. A list of economists should have been added in the database.
Postcondition	Owner is redirected to the Dashboard page.

Use Case 6

Name	View dashboard
Actor(s)	Owner
Description	Owner accesses economic reports, detailed farm information, and weather.
Precondition	Owner must be logged in the system.
Postcondition	Owner is redirected to Dashboard page.

Use Case 7

Name	Edit farm information
Actor(s)	Owner
Description	Owner clicks the Edit button to update any existing data. Owner clicks the Save button to confirm the changes done.
Precondition	Owner must be logged in the system. Data about the farm must be already stored in the database.
Postcondition	Owner is redirected to Dashboard page.

Use Case 8

Name	Manage crops
Actor(s)	Manager
Description	Manager enters the Manage Crops page. Manager can either add new crops or update existing crops' data. Manager clicks the Save button to confirm the changes done. Data are stored or updated in the database.
Precondition	Manager must be logged in the system. Data about crops must be already stored in the database.
Postcondition	Manager is redirected to Crops page.

Use Case 9

Name	Assign tasks
Actor(s)	Manager
Description	Manager assigns tasks to farm employees, including task duration and any other requirements.
Precondition	Manager must be logged in the system. A list of registered employees must already be stored in the database.
Postcondition	Manager is redirected to Dashboard page.

Use Case 10

Name	Manage livestock

Actor(s)	Manager
Description	Manager enters the Manage Livestock page. Manager can either add new livestock or update existing livestock's data. Manager clicks the Save button to confirm the changes done. Data are stored or updated in the database.
Precondition	Manager must be logged in the system. Data about crops must be already stored in the database.
Postcondition	Manager is redirected to Livestock page.

Use Case 11

Name	Manage farm information
Actor(s)	Manager
Description	Manager clicks the Edit button to update any existing data related to the farm. Manager clicks the Save button to confirm the changes done.
Precondition	Manager must be logged in the system. Information about the farm must be already stored in the database.
Postcondition	Manager is redirected to the Farm Information page.

Use Case 12

Name	Manage market
Actor(s)	Manager
Description	Manager enters the Manage Market page from the navigation bar. Manager can add buyers and sellers in the system, update information of existing buyers and sellers, or remove them from the database.
Precondition	Manager must be logged in the system.
Postcondition	Manager is redirected to the Dashboard page.

Use Case 13

Name	Manage veterinaries

Actor(s)	Manager
Description	Manager enters the Manage Users page. Manager can add new vets, edit existing data of a vet, or remove a vet from the system. The database is updated after this operation.
Precondition	Manager must be logged in the system. A list of registered veterinaries must be already stored in the database.
Postcondition	Manager is redirected to the Veterinaries page.

Use Case 14

Name	Manage other users
Actor(s)	Manager
Description	Manager enters the Manage Users page. Manager can add new users, edit existing data of a user, or remove a user from the system. The database is updated after this operation.
Precondition	Manager must be logged in the system. A list of existing users must be already stored in the database.
Postcondition	Manager is redirected to the Dashboard page.

Use Case 15

Name	Calculate profit
Actor(s)	Economist
Description	Economist performs calculations of the total farm profit (monthly and/or annual).
Precondition	Economist must be logged in the system. Previous information about profits in each farm sector must be available in the database.
Postcondition	Economist is redirected to the Dashboard page.

Use Case 16

Name	Create economic report
Actor(s)	Economist
Description	Economist completes a form after which a report is automatically generated.

Precondition	Economist must be logged in the system. Information related to economic aspects must be already stored in the database.
Postcondition	Economist is redirected to the Dashboard page.

Use Case 17

Name	Manage wages
Actor(s)	Economist
Description	Economist distributes wages to employees. Notifications are sent to employees.
Precondition	Economist must be logged in the system. A list of registered employees must be already stored in the database.
Postcondition	Economist is redirected to the Dashboard page.

Use Case 18

Name	View tasks
Actor(s)	Employee
Description	Employee checks tasks to be performed, assigned by the farm manager daily.
Precondition	Employee must be logged in the system.
Postcondition	Employee is redirected to the Home page.

Use Case 19

Name	Mark tasks as done
Actor(s)	Employee
Description	Employee checks completed tasks on the to-do list at the end of the day. Employee clicks the Save button to confirm the changes done.
Precondition	Employee must be logged in the system. A list of daily tasks must be accessible.
Postcondition	Employee is redirected to the Home page.

4.2 User Scenarios

4.2.1 Employee Scenarios

1. Employee Login

a. Employee enters ID and password.

b. If ID and password are correct and match the database, they are redirected to the employee homepage.

c. Homepage displays *Tasks Assigned* table and can display *Accomplished Tasks* table with a push of a button.

2. Employee Login fails

a. Employee enters ID and password.

b. If they do not match in the database an error will be shown letting them know something is incorrect.

3. Employee Views Personal Info

a. Employee clicks the profile picture or the username or the *Profile* link in dropdown menu.

b. Employee is redirected to a page showing his/her credentials like name, surname, age, phone number, wage, SSN, etc.

4. Employee Edits Personal Info

a. Employee clicks the *Settings* icon in the dropdown menu.

b. Employee enters new info (password, username, phone number).

c. Employee hits *Save* button and is required to enter his/her password as a security measurement.

d. Employees cannot change salary and SSN.

e. The data is updated and saved in database.

f. A message is shown to the employee to confirm that the data was changed and new data is being displayed in the profile page.

5. Employee Views Tasks

a. Tasks are shown in *Employee Homepage*, if no tasks a message informing the user there are no tasks required from him at this moment will be displayed.

b. If a task is clicked it shows a popup window displaying additional information with a *Mark as Done* button at the bottom.

c. Employee can sort tasks and choose to display up to 100 tasks at a time.

d. Tasks are shown in a table.

6. Employee Marks Tasks as Done

a. Employee clicks *Mark as Done* button in the task.

b. A task report is shown to be filled with a save button.

c. Task is put in *Accomplished Tasks* table.

d. A congratulation message is shown (not as a pop-up).

7. Employee Logout

a. Employee clicks the *Logout* button in their dropdown menu.

b. Employee can freely logout from account.

4.2.2 Owner Scenarios

8. Owner Logs In

a. Owner enters ID and password.

b. He/she is redirected to *Dashboard* page.

9. Owner Login fails

a. Owner enters ID and password.

b. If they do not match in the database an error will be shown letting them know something is incorrect.

10. Owner Views Personal Information

a. By clicking the profile picture, username or the profile link in the dropdown menu, owner is redirected to *Owner Information Page*.

b. Owner views his/her information.

11. Owner Edits Personal Info

a. Owner clicks the *Settings* icon in dropdown menu.

b. Owner is redirected to *Edit Information* page.

c. Owner changes his/her info (password, farm name, username, etc.).

d. Owner clicks *Save* button.

e. For security purposes, the system asks for confirmation and displays a message.

f. Database is updated after owner clicks *Yes* in the dialog box.

g. Owner is redirected to the same page now showing new information.

12. Owner Adds a Manager / Economist

a. Owner clicks the *Manage Farm* button in the dropdown menu.

b. Redirection to *Manage* page where 2 tables may be shown, *Manager* table and *Economist* table.

c. Owner clicks *Add New Manager* button to be redirected in a form page.

d. Owner fills the necessary data and confirms the new manager.

e. A random password (may be left as default) is generated and shown together with a *Send Mail* button.

f. Database is updated simultaneously.

13. Owner Removes a Manager / Economist

a. Owner goes to *Manage* page.

b. Owner clicks *Delete* on a manager or economist table entry.

c. A message showing a warning is displayed as a dialog box.
d. If owner chooses yes, the entry is erased from the database.

14. Owner Views Dashboard

a. After login *Owner* is redirected to owner homepage showing the *Dashboard*.
b. Owner can choose between *Crops*, *Livestock*, *Market*, *Vets*, *Weather*, and *Economic Report*.
c. Owner has only read-only rights.

15. Owner Edits Farm Information

a. Owner clicks *Edit Farm* Info in dropdown menu.
b. A page where all information that can be editable by owner in the farm is shown.
c. Owner makes changes and clicks *Save*.
d. A message is shown asking for conformation.
e. Database is updated.
f. New info is shown.

16. Owner Logs Out

a. Employee clicks the *Logout* button in their dropdown menu.
b. Employee can freely logout from account.

4.2.3 Economist Scenarios

17. Economist Logs In

a. Economist enters ID and password.
b. He/she is redirected to Dashboard page.

18. Economist Login fails

a. Economist enters ID and password.
b. If they do not match in the database an error will be shown letting them know something is incorrect.

19. Economist Views Personal Information

a. By clicking the profile picture, username or the profile link in the dropdown menu, economist is redirected to *Economist Information Page*.
b. Economist views his/her information.

20. Economist Edits Personal Info

a. Economist clicks the *Settings* icon in dropdown menu.
b. Economist is redirected to *Edit Information* page.
c. Economist changes his/her info (password, username, etc.).
d. For security purposes, the system asks for confirmation and displays a message.

e. Database is updated after economist clicks *Yes* in the dialog box.

f. Economist is redirected to the same page now showing new information.

21. Economist Calculates Profit

a. Economist goes to *Account* page where revenues and expenses for each sector/element are listed.

b. Economist chooses the time interval for which he/she will calculate the total revenue and clicks *Calculate Profit* button.

c. A popup box displays the calculated profit.

d. Database is updated after the calculation is performed.

e. Economist is redirected to the same page as before.

22. Economist Creates Economic Report

a. Economist goes to *Create Report* page.

b. Economist fills in the form with the required details to complete an economic report.

c. Database is updated after economist clicks the *Save* button at the end of the form.

d. Economist is redirected to the *Dashboard* page.

23. Economist Manages Wages

a. Economist goes to *Employees* page.

b. Economist uses a filtered search to find the employee.

c. Economist enters the page of a specific employee.

d. Economist assigns the monthly payment to the employee (and a bonus amount if available).

e. After the economist clicks the *Submit* button, a notification is sent to the employee about the payment transfer.

f. Database is updated.

g. Economist is redirected to *Employees* page.

24. Economist Logs Out

a. Economist clicks the *Logout* button in their dropdown menu.

b. Economist can freely logout from account.

4.2.4 Manager Scenarios

25. Manager Logs In

a. Manager enters ID and password.

b. He/she is redirected to Dashboard page.

26. Manager Login fails

a. Manager enters ID and password.

b. If they do not match in the database an error will be shown letting them know something is incorrect.

27. Manager Views Personal Information

a. By clicking the profile picture, username or the profile link in the dropdown menu, manager is redirected to *Manager Information* page.
b. Manager views his/her information.

28. Manager Edits Personal Info

a. Manager clicks the *Settings* icon in dropdown menu.
b. Manager is redirected to *Edit Information* page.
c. Manager changes his/her info (password, username, etc.).
d. For security purposes, the system asks for confirmation and displays a message.
e. Database is updated after manager clicks *Yes* in the dialog box.
f. Manager is redirected to the same page now showing new information.

29. Manager Adds Crops

a. Manager enters the *Manage Crops* page from the *Dashboard* page.
b. Manager clicks *Add Crop* button and is redirected to another page.
c. Manager fills in the form and clicks *Save* button.
d. For security purposes, the system asks for confirmation and displays a message.
e. Database is updated after manager clicks *Yes* in the dialog box.
f. Manager is redirected to the *Manage Crops* page.

30. Manager Edits Crops

a. Manager enters the *Manage Crops* page from the *Dashboard* page.
b. Manager clicks *Edit Crop* button and is redirected to another page.
c. Manager refills in the form and clicks *Save* button.
d. For security purposes, the system asks for confirmation and displays a message.
e. Database is updated after manager clicks *Yes* in the dialog box.
f. Manager is redirected to the *Manage Crops* page.

31. Manager Assigns Tasks

a. Manager goes to *Employees* page.
b. Manager uses a filtered search to find the employee.
c. Manager enters the page of a specific employee.
d. Manager assigns tasks to the employee. Task duration must not surpass the maximum amount of working hours in a day.
e. After the manager clicks the save button, a notification is sent to the employees about the tasks assignment.
f. Database is updated.
g. Economist is redirected to Employees page.

32. Manager Adds Livestock

a. Manager enters the *Manage Livestock* page from the *Dashboard* page.
b. Manager clicks *Add Livestock* button and is redirected to another page.
c. Manager fills in the form and clicks *Save* button.
d. For security purposes, the system asks for confirmation and displays a message.
e. Database is updated after manager clicks *Yes* in the dialog box.
f. Manager is redirected to the *Manage Livestock* page.

33. Manager Edits Livestock

a. Manager enters the *Manage Livestock* page.
b. Manager clicks *Edit Livestock* button and is redirected to another page.
c. Manager refills in the form and clicks *Save* button.
d. For security purposes, the system asks for confirmation and displays a message.
e. Database is updated after manager clicks *Yes* in the dialog box.
f. Manager is redirected to the *Manage Livestock* page.

34. Manager Manages Farm Information

a. Manager clicks *Edit Farm Information* in dropdown menu.
b. A page where all information that can be editable by owner in the farm is shown.
c. Owner makes changes and clicks *Save*.
d. A message is shown asking for conformation.
e. Database is updated.
f. New info is shown.

35. Manager Adds a Buyer / Supplier

a. Manager enters the *Market* page from the *Dashboard* page.
b. Manager clicks *Add New Buyer* or *Add New Supplier* button and is redirected to another page.
c. Manager fills in the form with buyer/supplier information and confirms the new buyer/supplier registration.
d. Database is updated simultaneously.

36. Manager Edits a Buyer / Supplier

a. Manager enters the *Market* page from the *Dashboard* page.
b. Manager uses filtered search to find a buyer or supplier.
c. Manager clicks *Edit* button for a buyer or supplier and is redirected to another page.
d. Manager fills in the new form and confirms the update.
e. Database is updated simultaneously.

37. Manager Removes a Buyer / Supplier

a. Manager enters the *Market* page from the *Dashboard* page.

b. Manager is redirected to another page.
c. Manager uses filtered search to find a buyer or supplier.
d. Manager clicks *Delete* on the buyer's or supplier's table entry.
e. A message showing a warning is displayed as a dialog box.
f. If manager chooses *Yes*, the entry is erased from the database.

38. Manager Adds a Veterinary / Other User

a. Manager clicks the *Manage Vets* or *Manage Other Users* button in the *Dashboard* page.
b. Manager is redirected to another page.
c. Manager clicks *Add New Vet* or *Add New User* button and is redirected to another page.
d. Manager fills in the form with vet/user information and confirms the new vet/user registration.
e. A random password (may be left as default) is generated and shown together with a *Send Mail* button.
f. Database is updated simultaneously.

39. Manager Edits a Veterinary / Other User

a. Manager clicks the *Manage Vets* or *Manage Other Users* button in the *Dashboard* page.
b. Manager is redirected to another page.
c. Manager uses filtered search to find a vet or user.
d. Manager clicks *Edit* button for a vet or user and is redirected to another page.
e. Manager fills in the new form and confirms the update.
f. Database is updated simultaneously.

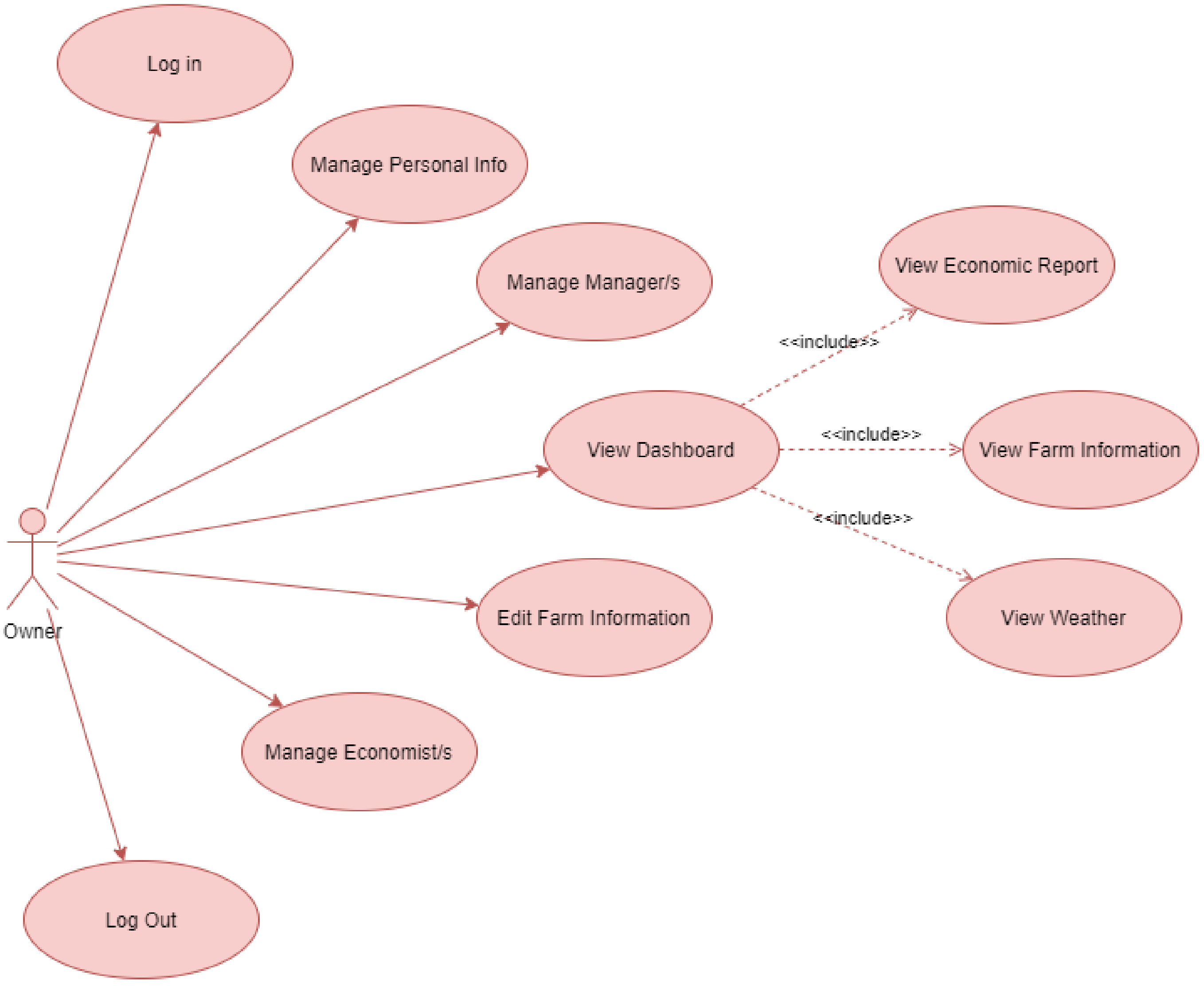
40. Manager Removes a Veterinary / Other User

a. Manager clicks the *Manage Vets* or *Manage Other Users* button in the *Dashboard* page.
b. Manager is redirected to another page.
c. Manager uses filtered search to find a vet or user.
d. Manager clicks *Delete* on the vet's or user's table entry.
e. A message showing a warning is displayed as a dialog box.
f. If manager chooses *Yes*, the entry is erased from the database.

4.3 Use Case Diagrams

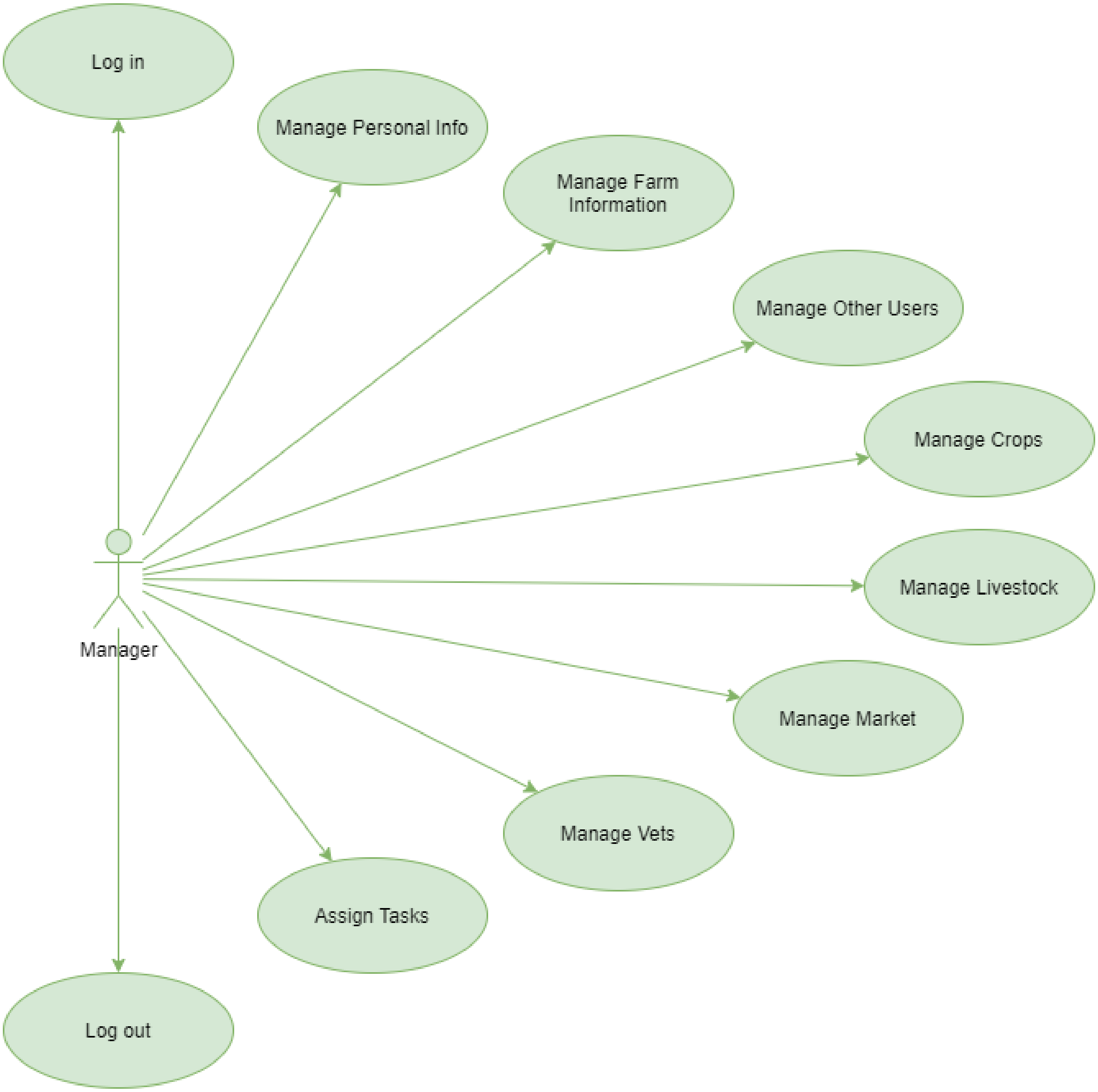
4.3.1 Farm Owner

Click here for a full view: Owner Use Case Diagram.



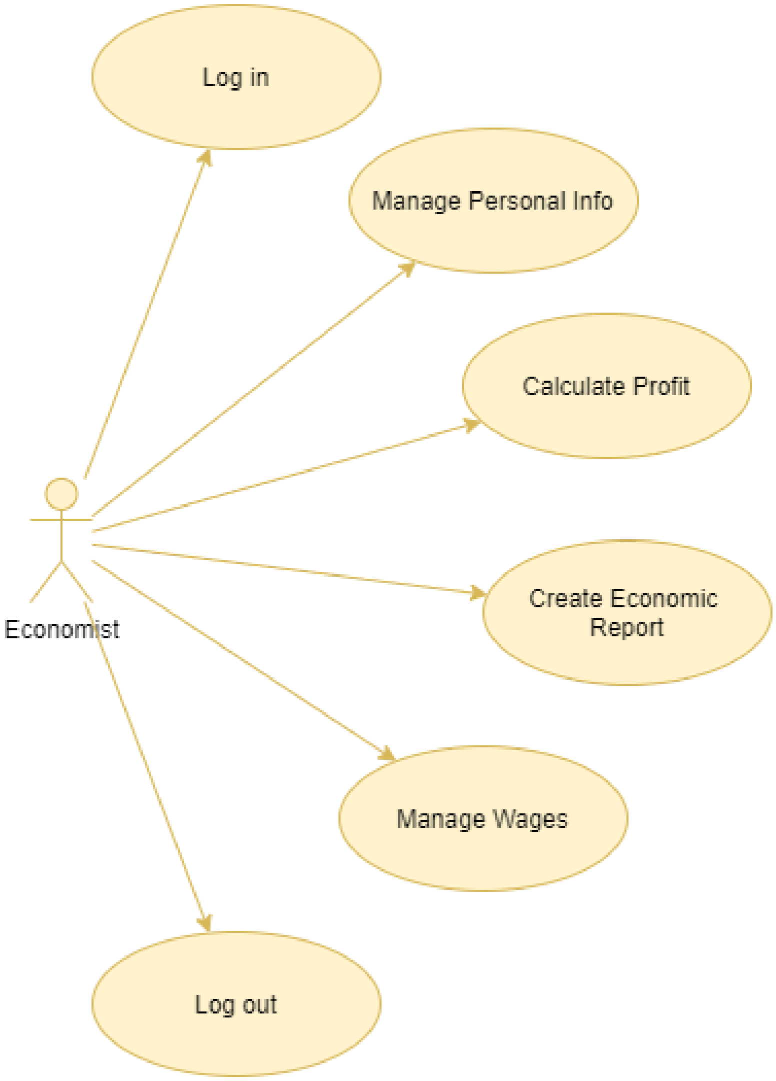
4.3.2 Farm Manager

Click here for a full view: Manager Use Case Diagram.



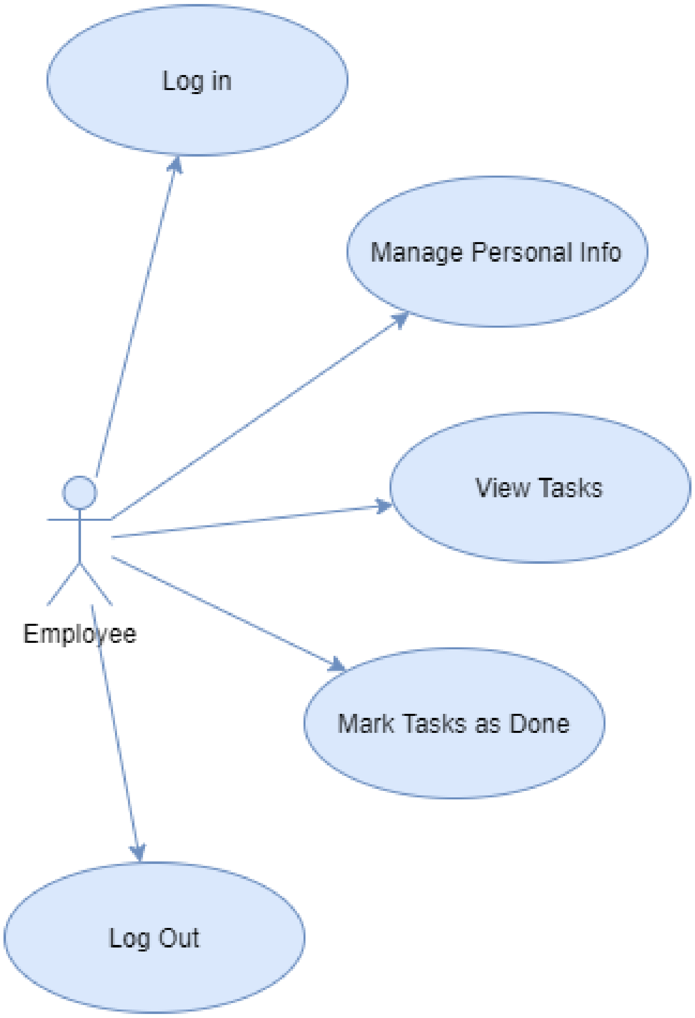
4.3.3 Farm Economist

Click here for a full view: Economist Use Case Diagram.



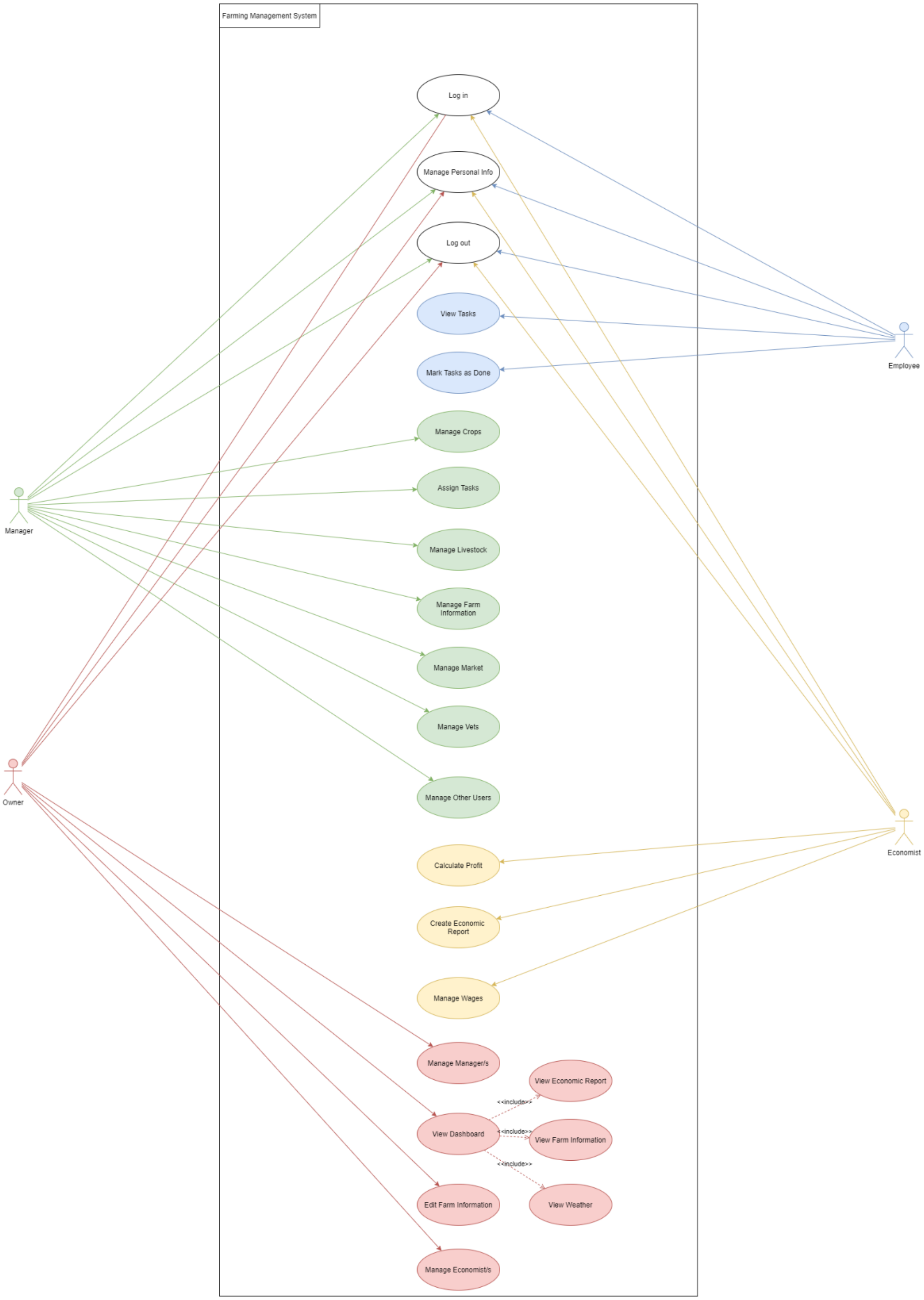
4.3.4 Farm Employee

Click here for a full view: Employee Use Case Diagram.



4.3.5 Full Use Case Diagram

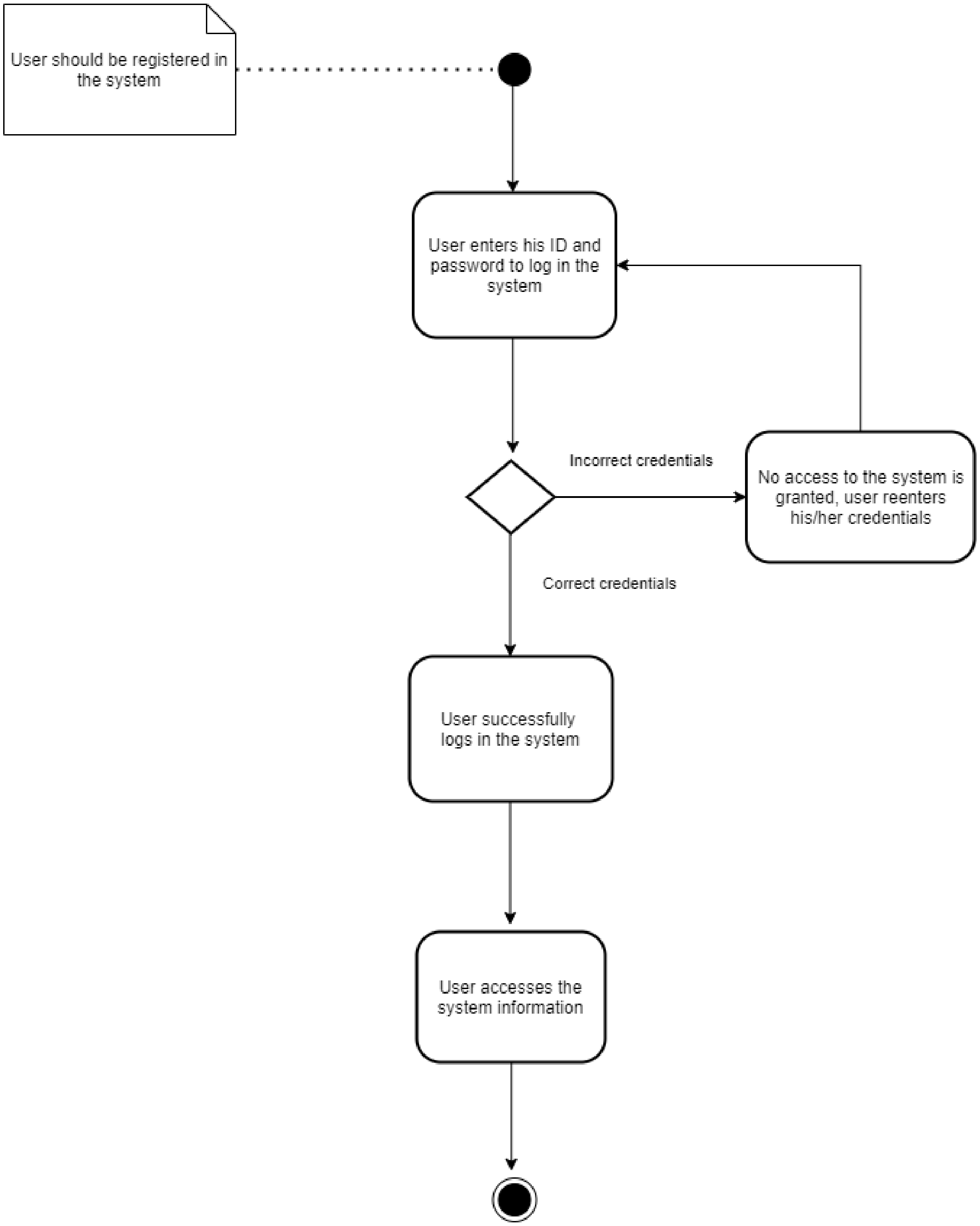
Click here for a full view: Full Use Case Diagram.



4.4 Activity Diagrams

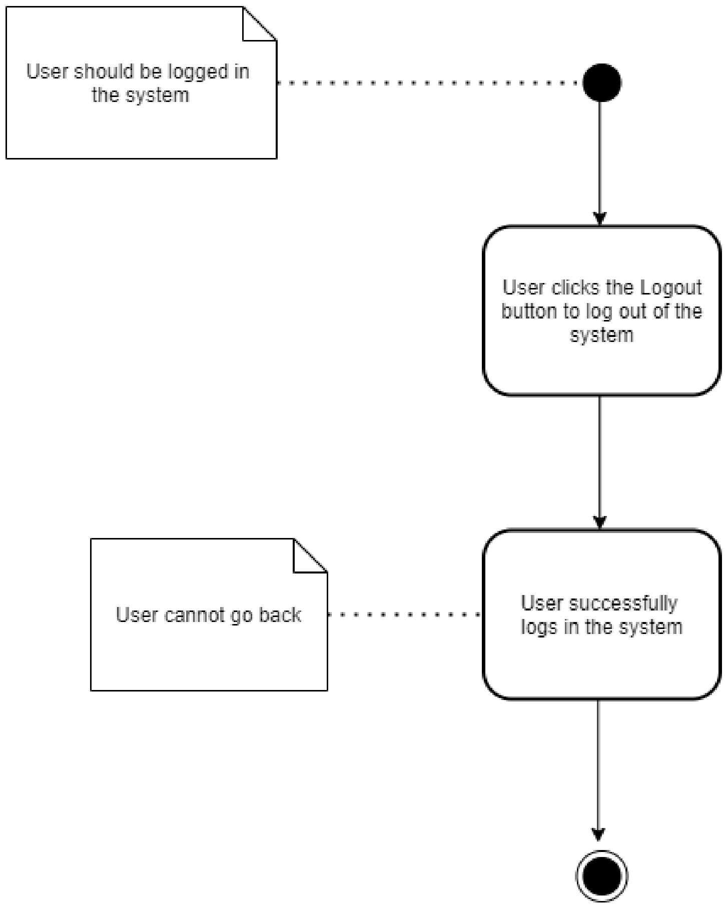
4.4.1 AC_01

Click here for a full view: AC_01.


User should be registered in the system
User enters his ID and password to log in the system
Incorrect credentials
No access to the system is granted, user reenters his/her credentials
Correct credentials
User successfully logs in the system
User accesses the system information

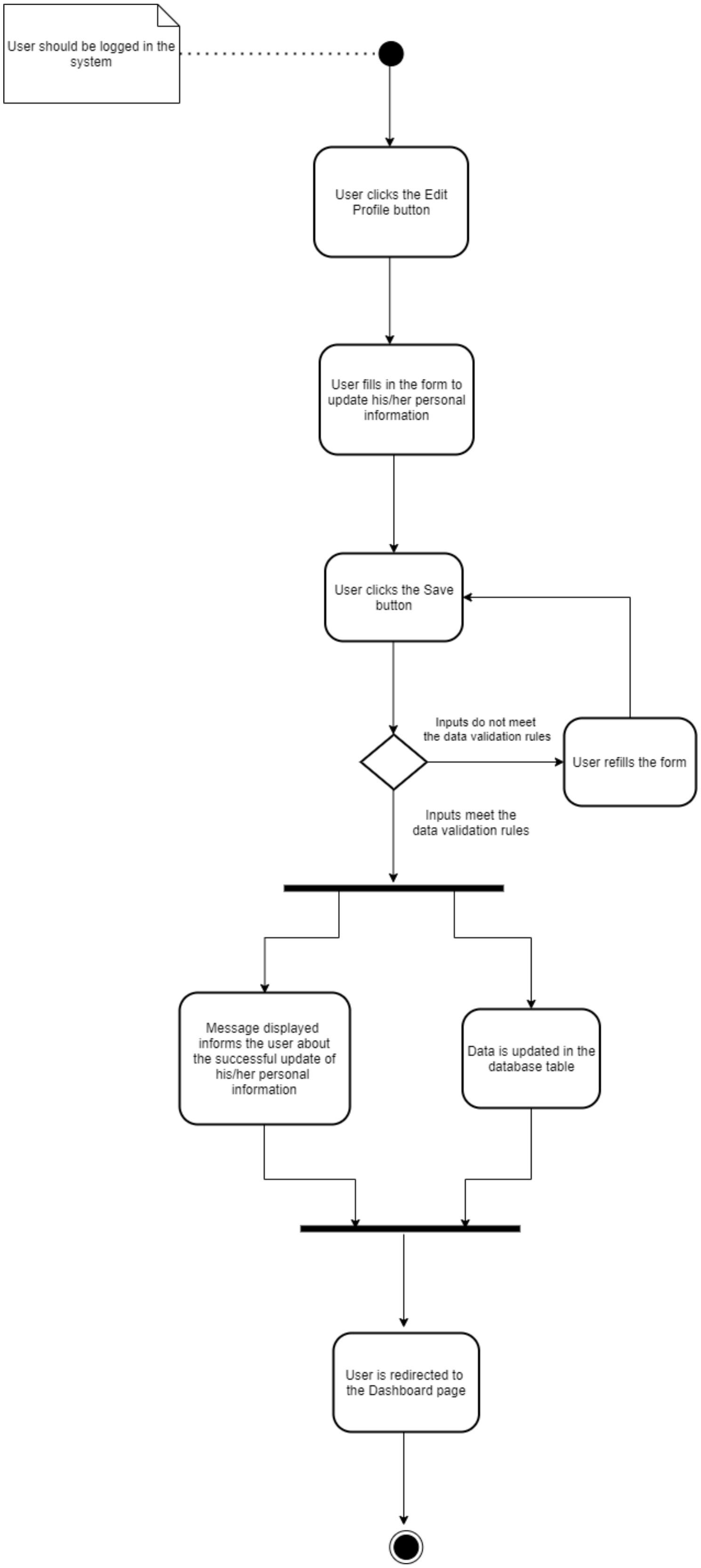
4.4.2 AC_02

Click here for a full view: AC_02.


User should be logged in the system
User clicks the Logout button to log out of the system
User cannot go back
User successfully logs in the system

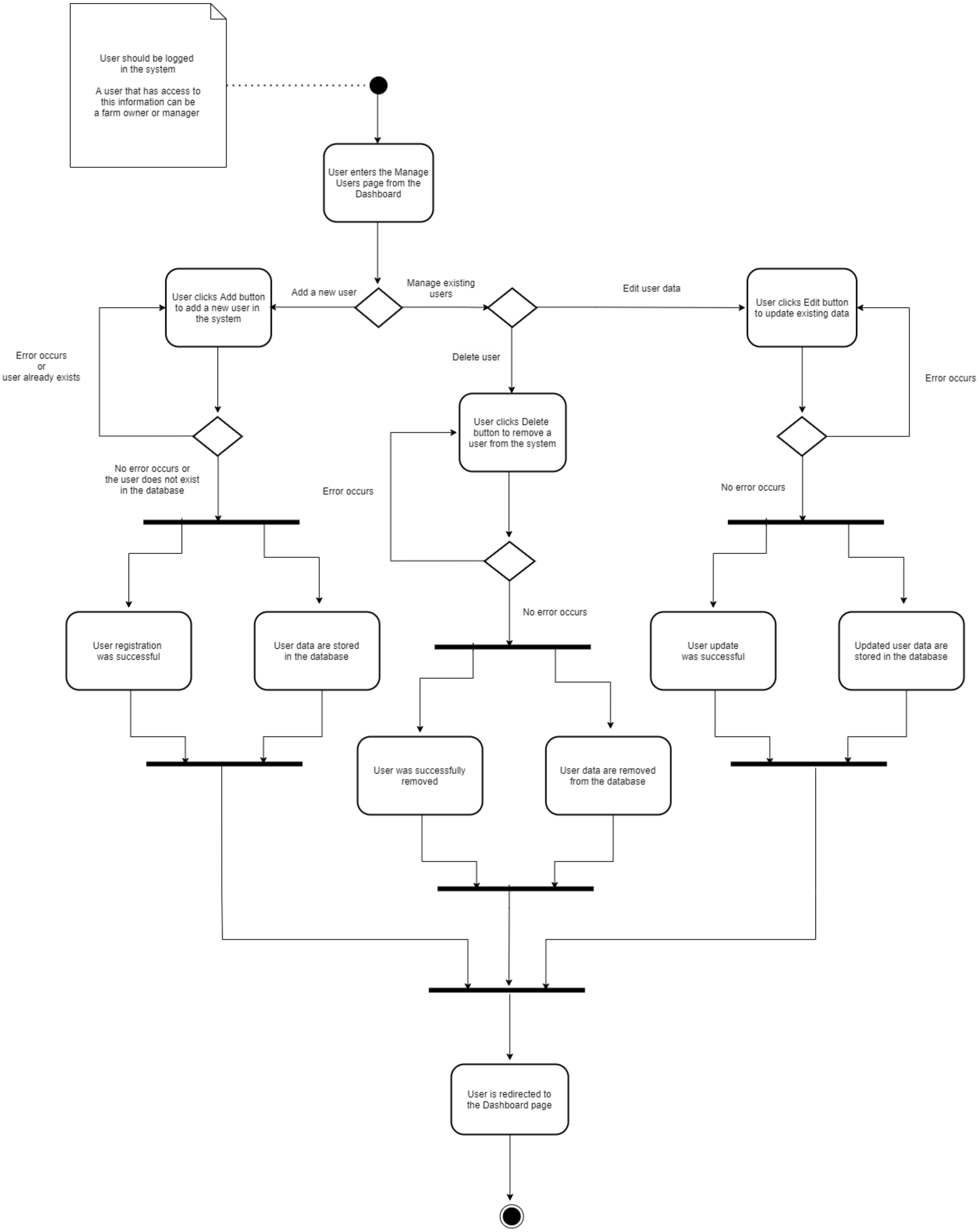
4.4.3 AC_03

Click here for a full view: AC_03.


User should be logged in the system
User clicks the Edit Profile button
User fills in the form to update his/her personal information
User clicks the Save button
Inputs do not meet the data validation rules
User refills the form
Inputs meet the data validation rules
Message displayed informs the user about the successful update of his/her personal information
Data is updated in the database table
User is redirected to the Dashboard page

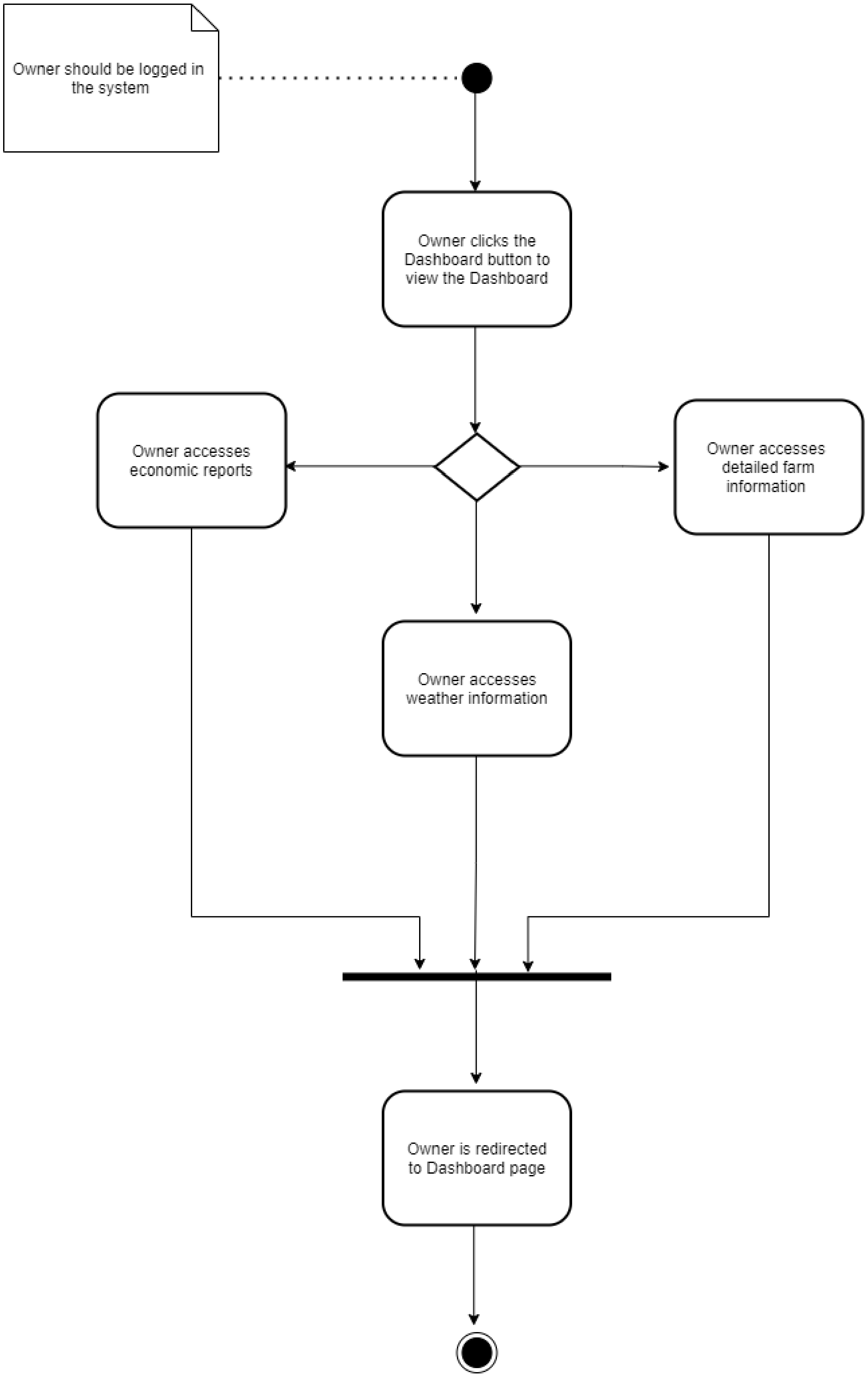
4.4.4 AC_04_05_13_14

Click here for a full view: AC_04_05_13_14.


User should be logged in the system
A user that has access to this information can be a farm owner or manager
User enters the Manage Users page from the Dashboard
Add a new user
Manage existing users
Edit user data
User clicks Add button to add a new user in the system
User clicks Edit button to update existing data
Error occurs or user already exists
Delete user
Error occurs
User clicks Delete button to remove a user from the system
No error occurs or the user does not exist in the database
Error occurs
No error occurs
No error occurs
User registration was successful
User data are stored in the database
User update was successful
Updated user data are stored in the database
User was successfully removed
User data are removed from the database
User is redirected to the Dashboard page

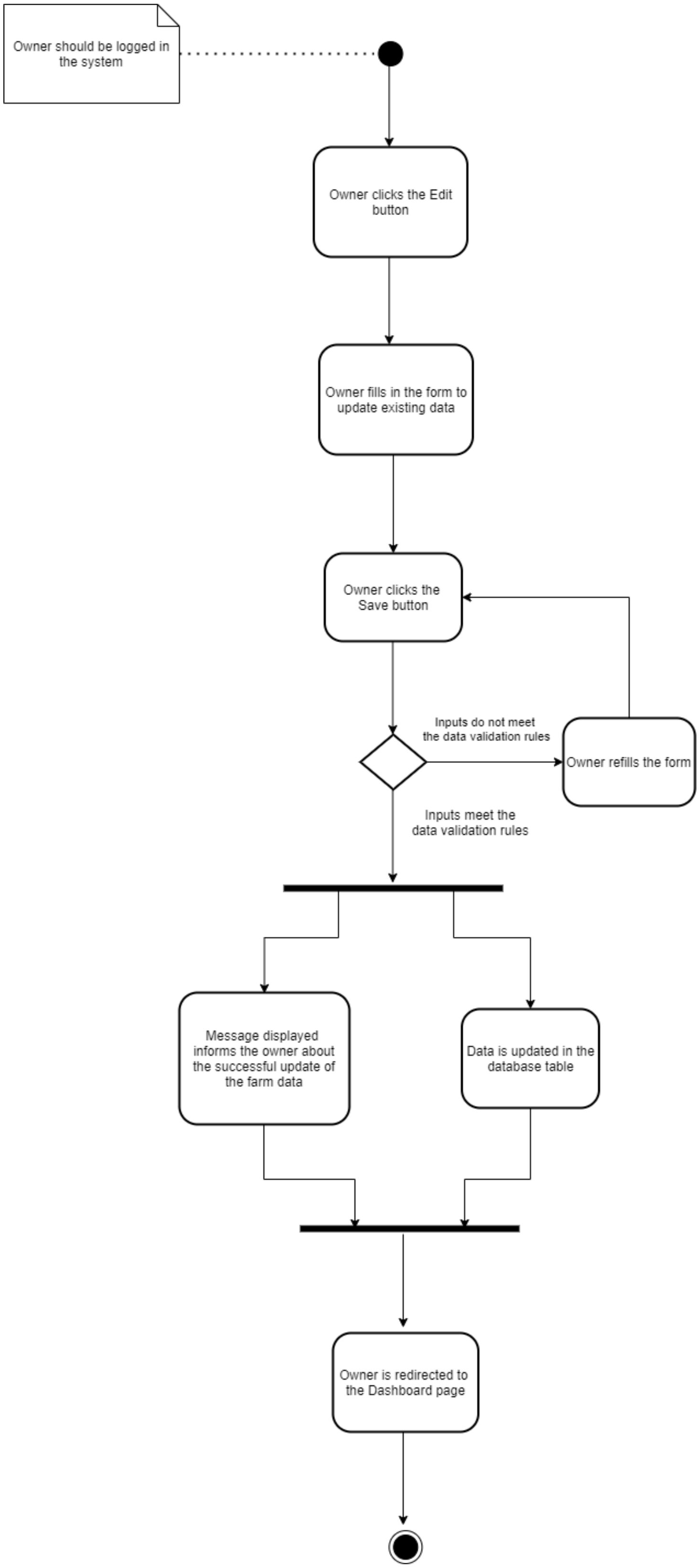
4.4.5 AC_06

Click here for a full view: AC_06.


Owner should be logged in the system
Owner clicks the Dashboard button to view the Dashboard
Owner accesses economic reports
Owner accesses detailed farm information
Owner accesses weather information
Owner is redirected to Dashboard page

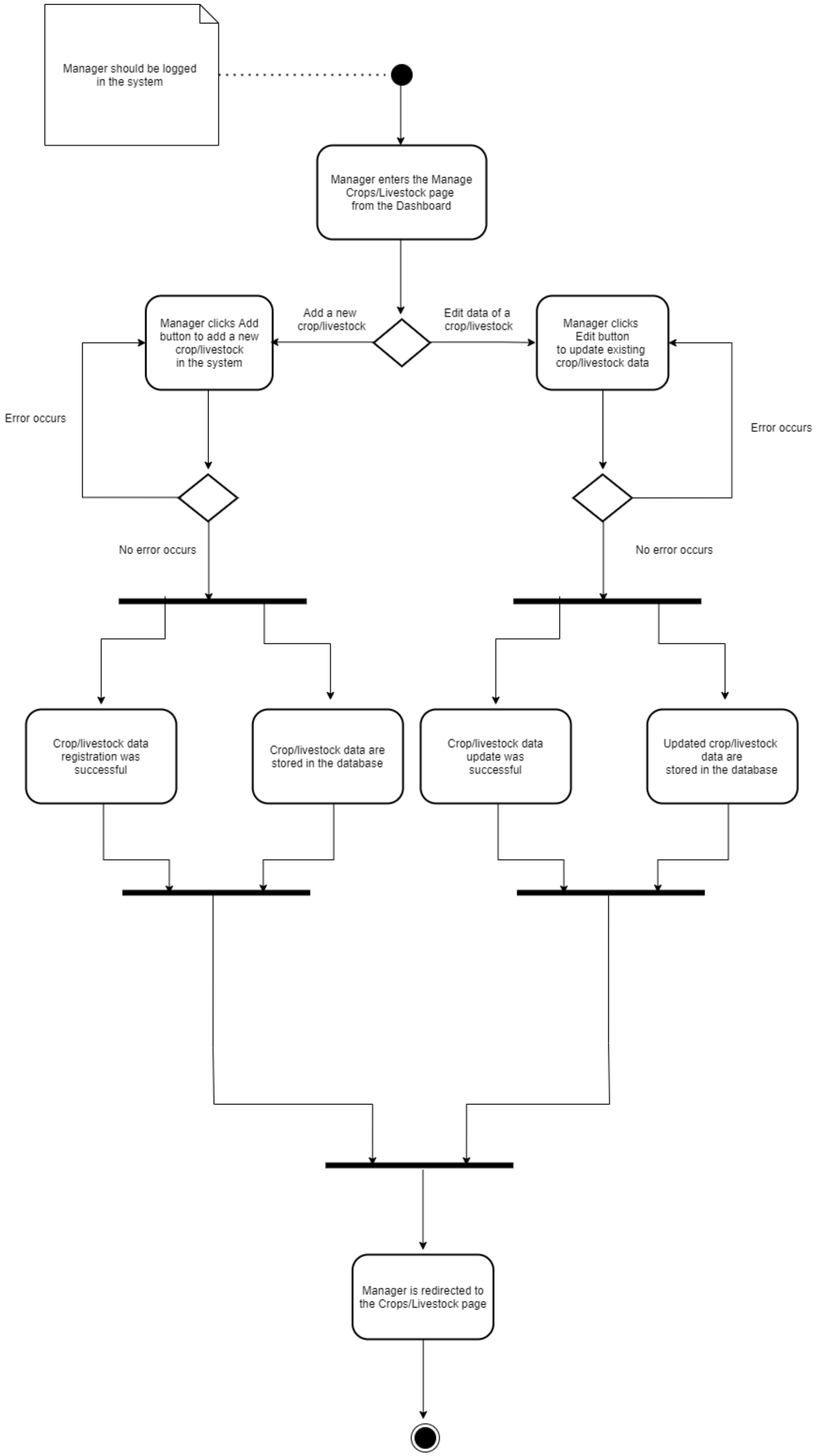
4.4.6 AC_07_11

Click here for a full view: AC_07_11.


Owner should be logged in the system
Owner clicks the Edit button
Owner fills in the form to update existing data
Owner clicks the Save button
Inputs do not meet the data validation rules
Owner refills the form
Inputs meet the data validation rules
Message displayed informs the owner about the successful update of the farm data
Data is updated in the database table
Owner is redirected to the Dashboard page

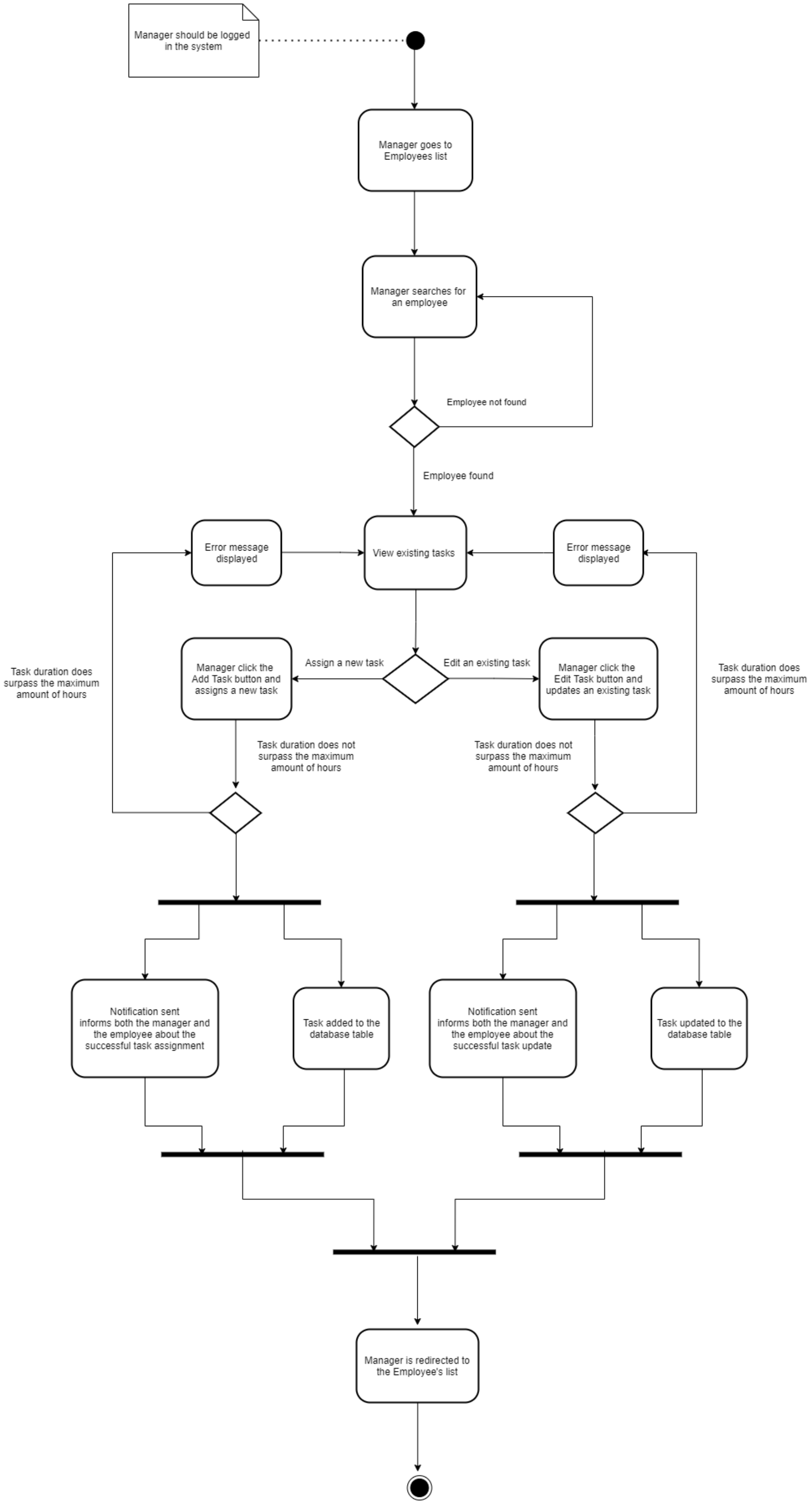
4.4.7 AC_08_10

Click here for a full view: AC_08_10.


Manager should be logged in the system
Manager enters the Manage Crops/Livestock page from the Dashboard
Add a new crop/livestock
Edit data of a crop/livestock
Manager clicks Add button to add a new crop/livestock in the system
Manager clicks Edit button to update existing crop/livestock data
Error occurs
Error occurs
No error occurs
No error occurs
Crop/livestock data registration was successful
Crop/livestock data are stored in the database
Crop/livestock data update was successful
Updated crop/livestock data are stored in the database
Manager is redirected to the Crops/Livestock page

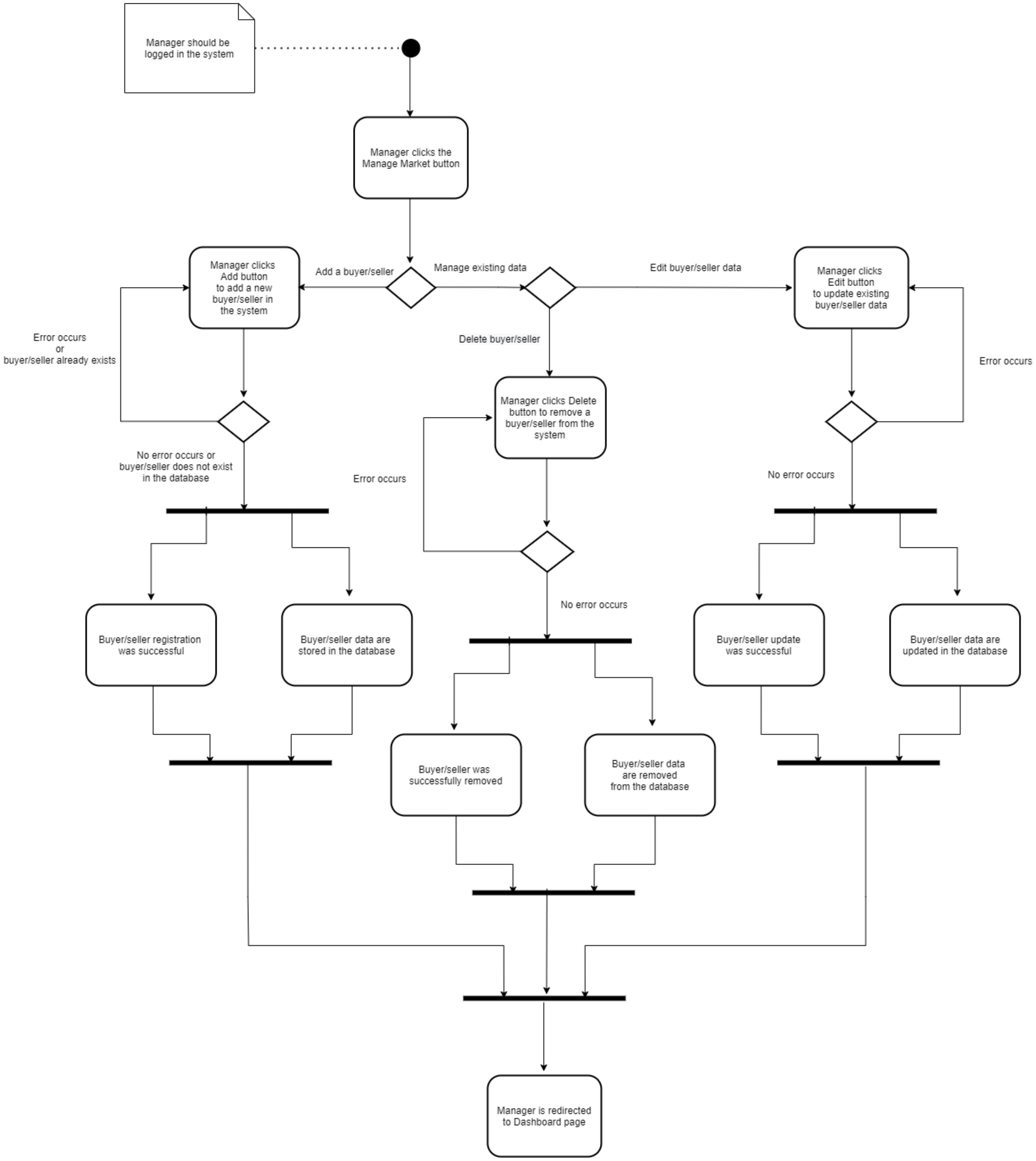
4.4.8 AC_09

Click here for a full view: AC_09.


Manager should be logged in the system
Manager goes to Employees list
Manager searches for an employee
Employee not found
Employee found
Error message displayed
View existing tasks
Error message displayed
Task duration does surpass the maximum amount of hours
Manager click the Add Task button and assigns a new task
Assign a new task
Edit an existing task
Manager click the Edit Task button and updates an existing task
Task duration does surpass the maximum amount of hours
Task duration does not surpass the maximum amount of hours
Task duration does not surpass the maximum amount of hours
Notification sent informs both the manager and the employee about the successful task assignment
Task added to the database table
Notification sent informs both the manager and the employee about the successful task update
Task updated to the database table
Manager is redirected to the Employee's list

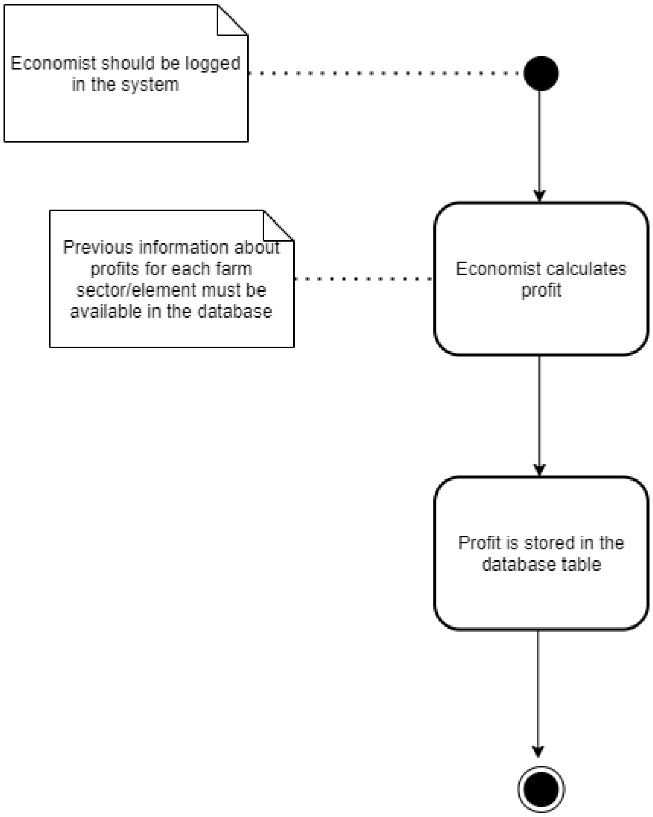
4.4.9 AC_12

Click here for a full view: AC_12.


Manager should be logged in the system
Manager clicks the Manage Market button
Add a buyer/seller
Manage existing data
Edit buyer/seller data
Manager clicks Add button to add a new buyer/seller in the system
Manager clicks Edit button to update existing buyer/seller data
Delete buyer/seller
Error occurs or buyer/seller already exists
Error occurs
Manager clicks Delete button to remove a buyer/seller from the system
No error occurs or buyer/seller does not exist in the database
Error occurs
No error occurs
No error occurs
Buyer/seller registration was successful
Buyer/seller data are stored in the database
Buyer/seller update was successful
Buyer/seller data are updated in the database
Buyer/seller was successfully removed
Buyer/seller data are removed from the database
Manager is redirected to Dashboard page

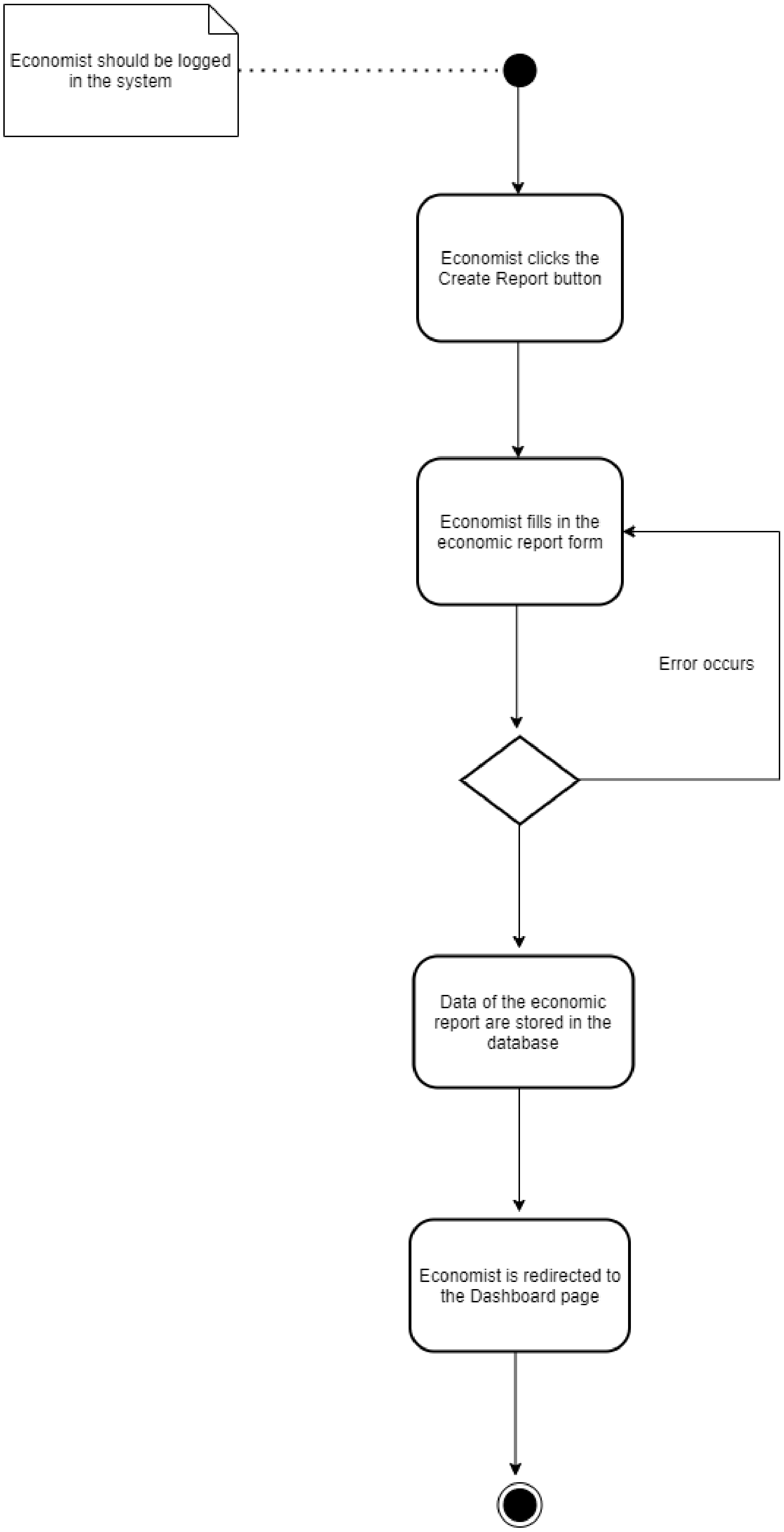
4.4.10 AC_15

Click here for a full view: AC_15.


Economist should be logged in the system
Previous information about profits for each farm sector/element must be available in the database
Economist calculates profit
Profit is stored in the database table

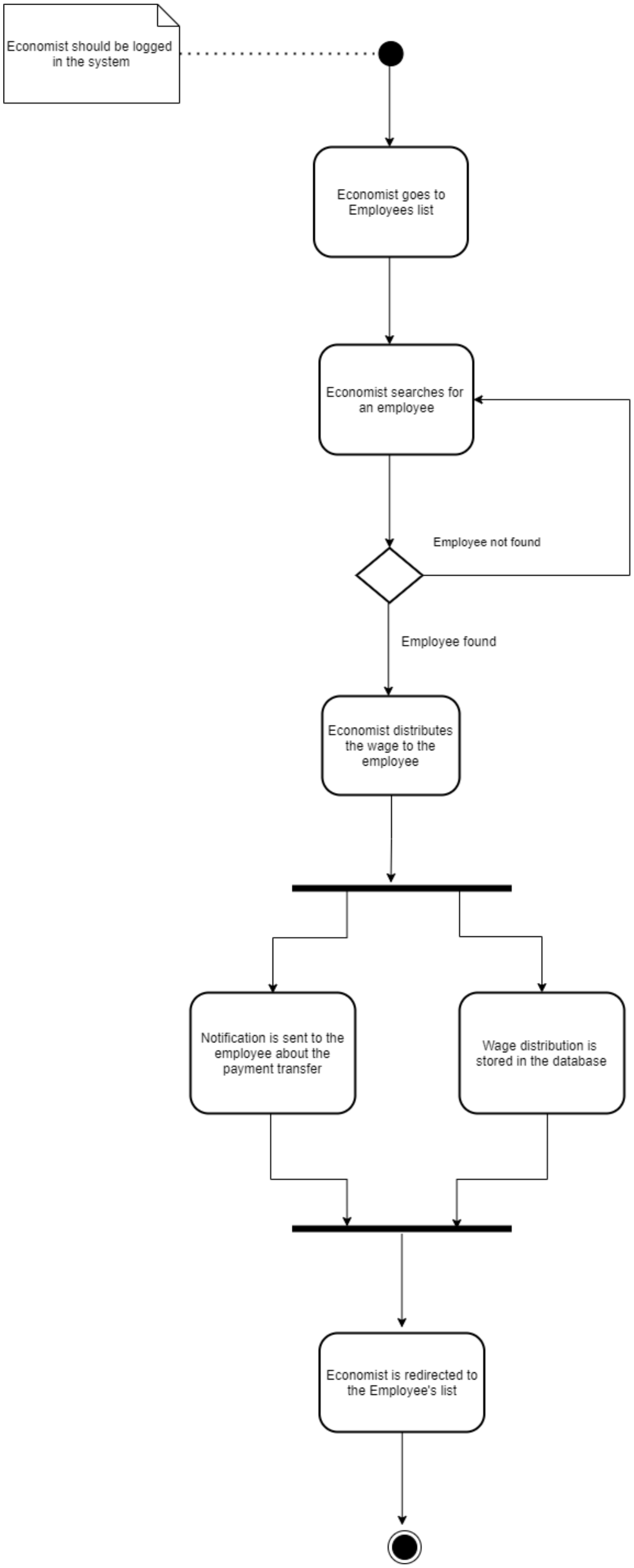
4.4.11 AC_16

Click here for a full view: AC_16.


Economist should be logged in the system
Economist clicks the Create Report button
Economist fills in the economic report form
Error occurs
Data of the economic report are stored in the database
Economist is redirected to the Dashboard page

4.4.12 AC_17

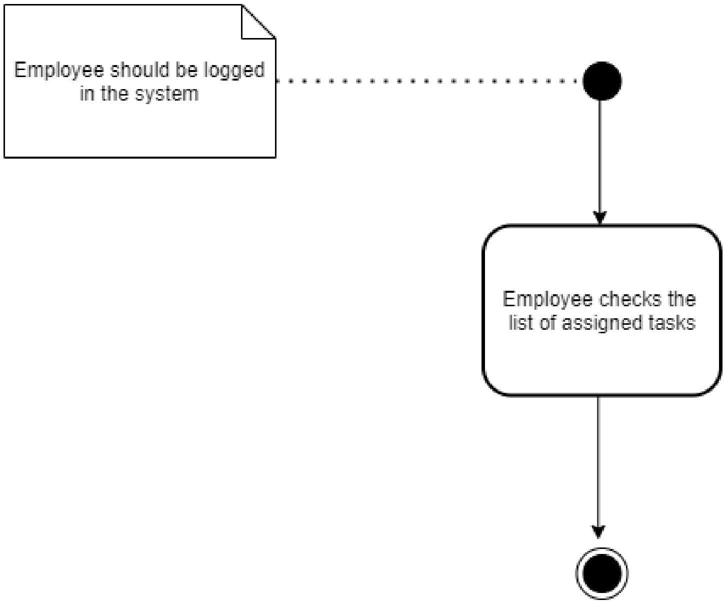
Click here for a full view: AC_17.



Economist should be logged in the system
Economist goes to Employees list
Economist searches for an employee
Employee not found
Employee found
Economist distributes the wage to the employee
Notification is sent to the employee about the payment transfer
Wage distribution is stored in the database
Economist is redirected to the Employee's list

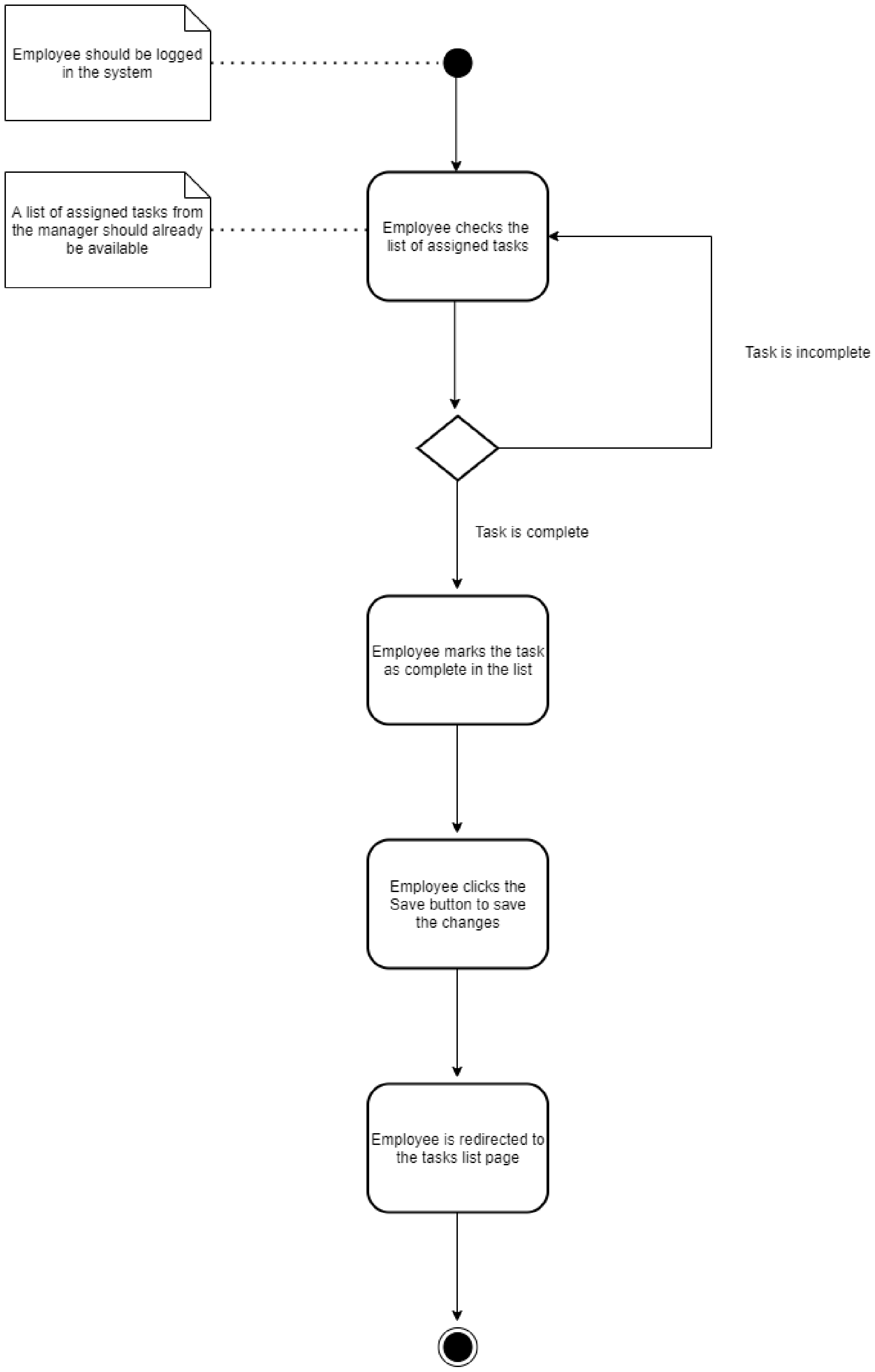
4.4.13 AC_18

Click here for a full view: AC_18.


Employee should be logged in the system
Employee checks the list of assigned tasks

4.4.14 AC_19

Click here for a full view: Full Use Case Diagram.


Employee should be logged in the system
A list of assigned tasks from the manager should already be available
Employee checks the list of assigned tasks
Task is incomplete
Task is complete
Employee marks the task as complete in the list
Employee clicks the Save button to save the changes
Employee is redirected to the tasks list page

4.5 State Diagrams

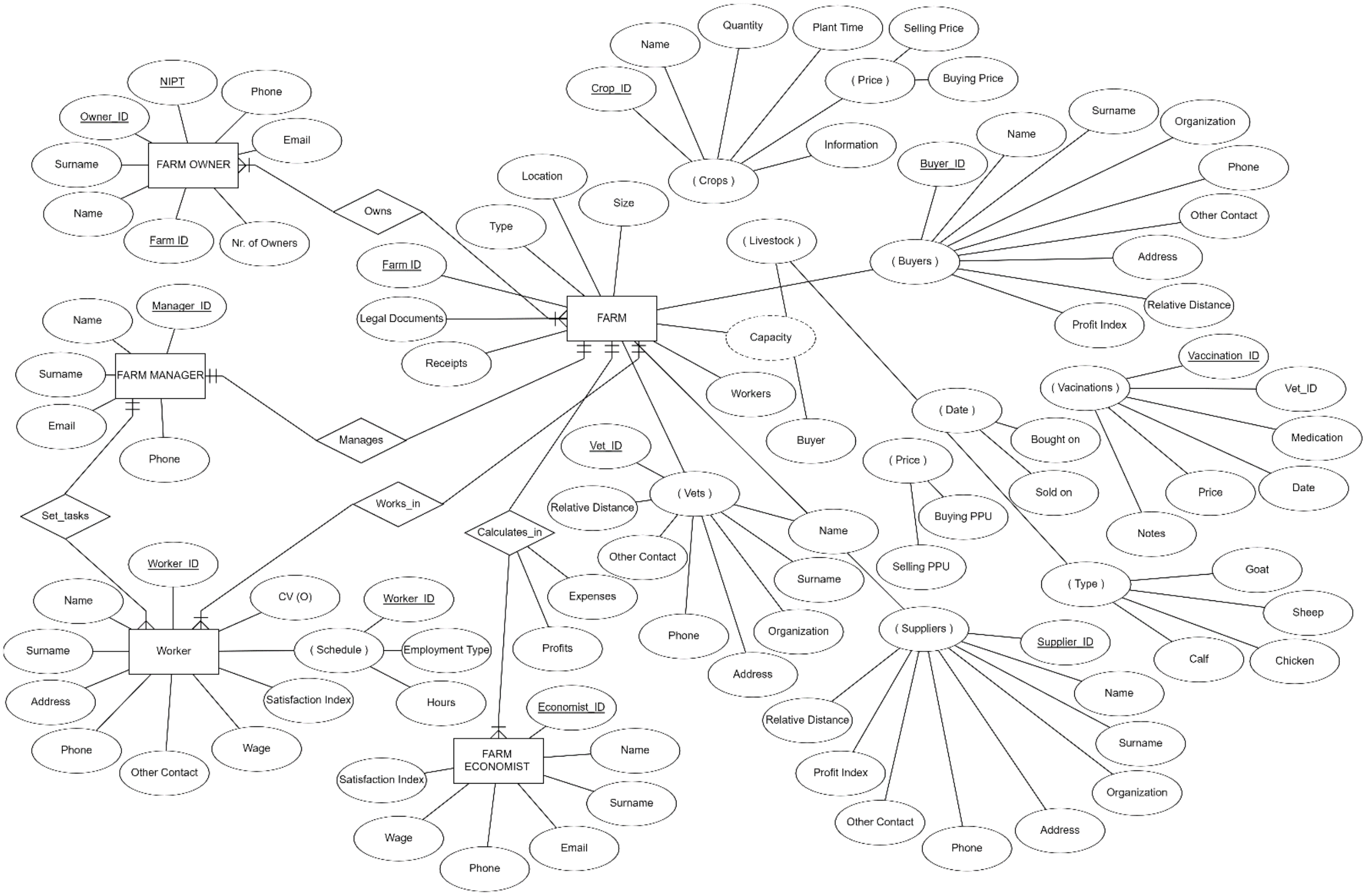
4.6 Sequence Diagrams

4.7 Collaboration Diagrams

4.8 Data Flow Diagrams

4.9 Entity Relationship Diagram

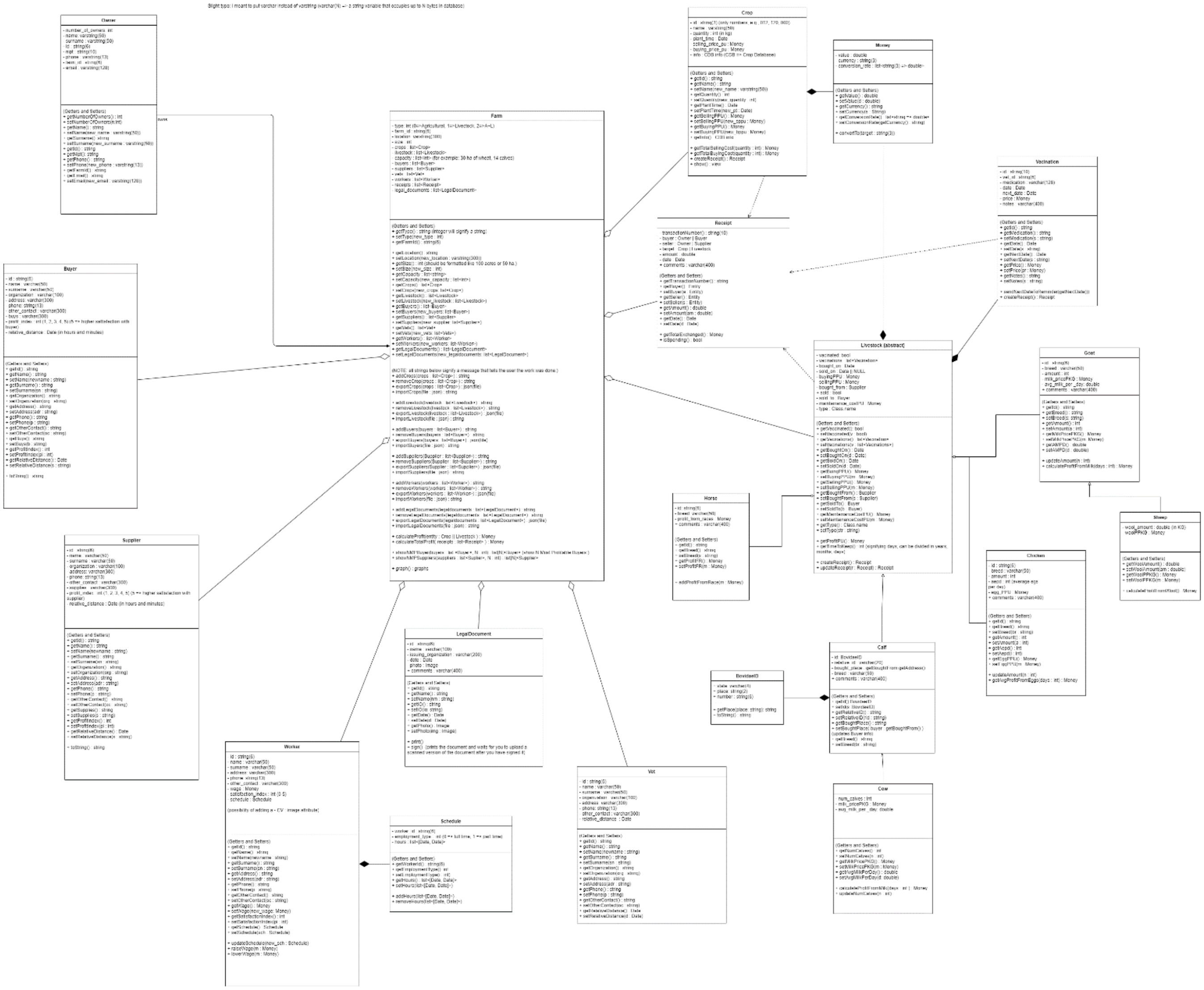
Click here for a full view: ER Diagram.



4.10 Relational Schema

4.11 Class Diagram

Click here for a full view: Class Diagram.



5. Implementation Technology

Our software is a web app built with the Laravel framework. In order to deploy the project from his/her own computer one would have to follow these steps:

5.1 Prerequisites

1. XAMPP web server solution stack 8+ (https://www.apachefriends.org/index.html)

- PHP 8.0.x which comes with XAMPP

- MariaDB fork of the MySQL relational database management system comes with XAMPP

- (optional) Apache web server also comes with XAMPP

2. Composer package manager for PHP (version 2.0.8) (https://getcomposer.org/download/)

3. (optional) npm package manager (version 7.0.8) which comes with Node.js (15.1.0) (https://nodejs.org/en/download/)

4. Git installed in your system (2.31.1) (https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)

5.2 Steps to implement in your own computer

1. Clone the repository with git clone https://github.com/kdomi18/kot.git

2. Go to the folder application using cd command on your cmd or terminal

3. Run composer install on your cmd or terminal

4. Copy .env.example file to .env on the root folder. You can type copy .env.example .env if using command prompt Windows or cp .env.example .env if using terminal, Ubuntu

5. Open your .env file and change the database name (DB_DATABASE) to whatever you have, username (DB_USERNAME) and password (DB_PASSWORD) field correspond to your configuration.

By default, the username is root, and you can leave the password field empty. (This is for Xampp)

By default, the username is root and password is also root. (This is for Lamp)

6. Start XAMPP Control Panel

7. Start MySQL sevice and Apache service with XAMPP control panel

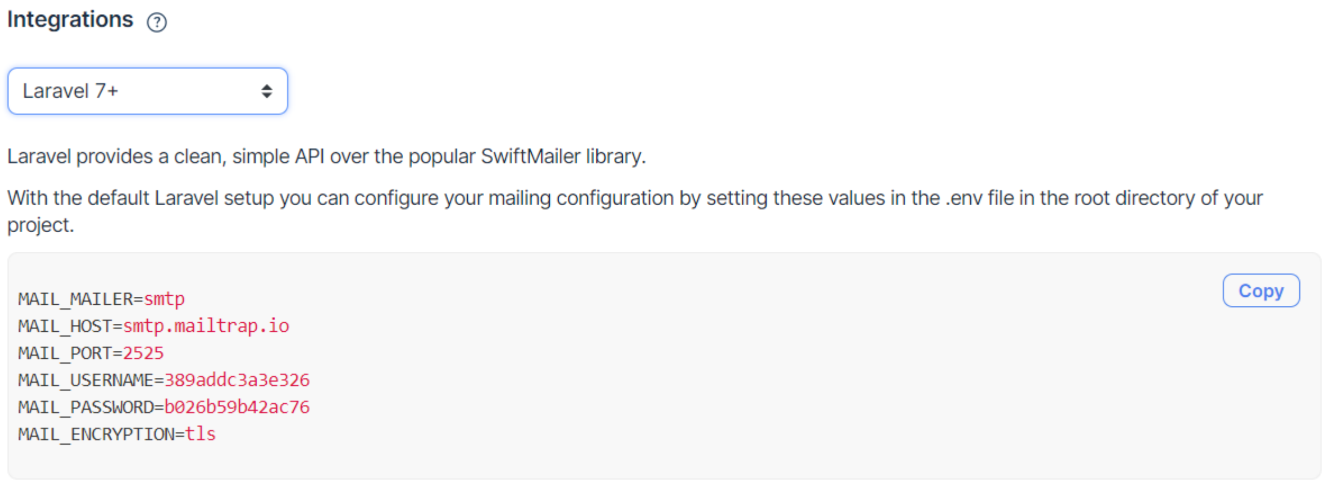
8. Run php artisan key:generate

9. Run php artisan migrate --seed

10. If you want to register a new account or be able to access User Management view you need to follow the following steps, else you may skip them.

   - Create a new account with https://mailtrap.io/

   - Choose Laravel 7+ integration like below

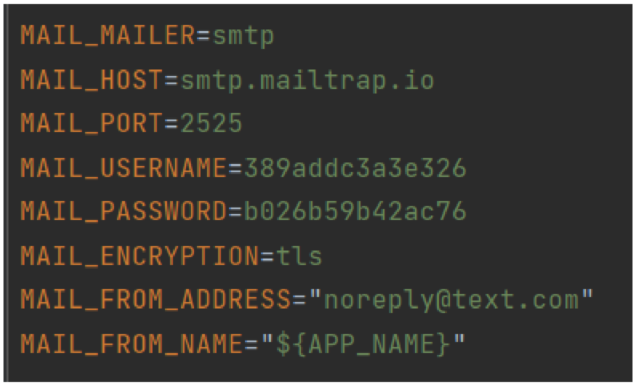


Laravel provides a clean, simple API over the popular SwiftMailer library.

With the default Laravel setup you can configure your mailing configuration by setting these values in the .env file in the root directory of your project.

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=389addc3a3e326
MAIL_PASSWORD=b026b59b42ac76
MAIL_ENCRYPTION=tls
```

   - Copy the 6 lines provided there and change the configuration in .env file like so:



```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=389addc3a3e326
MAIL_PASSWORD=b026b59b42ac76
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="noreply@text.com"
MAIL_FROM_NAME="${APP_NAME}"
```

- Run php artisan serve

- Go to http://127.0.0.1:8000

- Go to register page through top right button

- Follow the steps to register

- Click the reset password link sent with an email in your newly created mailtrap.io account

- Choose a password for yourself and log in

11. If you prefer not to register then log in with credentials email : admin@admin.com, password : password (NOTE: you'll need to set up mailtrap.io like I explained above in order to access User Management view)

These steps are also shown in the code repository: https://github.com/kdomi18/kot

6. Project Planning

Project Name: Farming Management System

Members: Endri Seferi, Anila Hoxha, Batjona Tahiraj, Alessia Toli, Arber Sadushi, Kejdi Domi

Start and end dates: 07 April 2021 – 15 June 2021

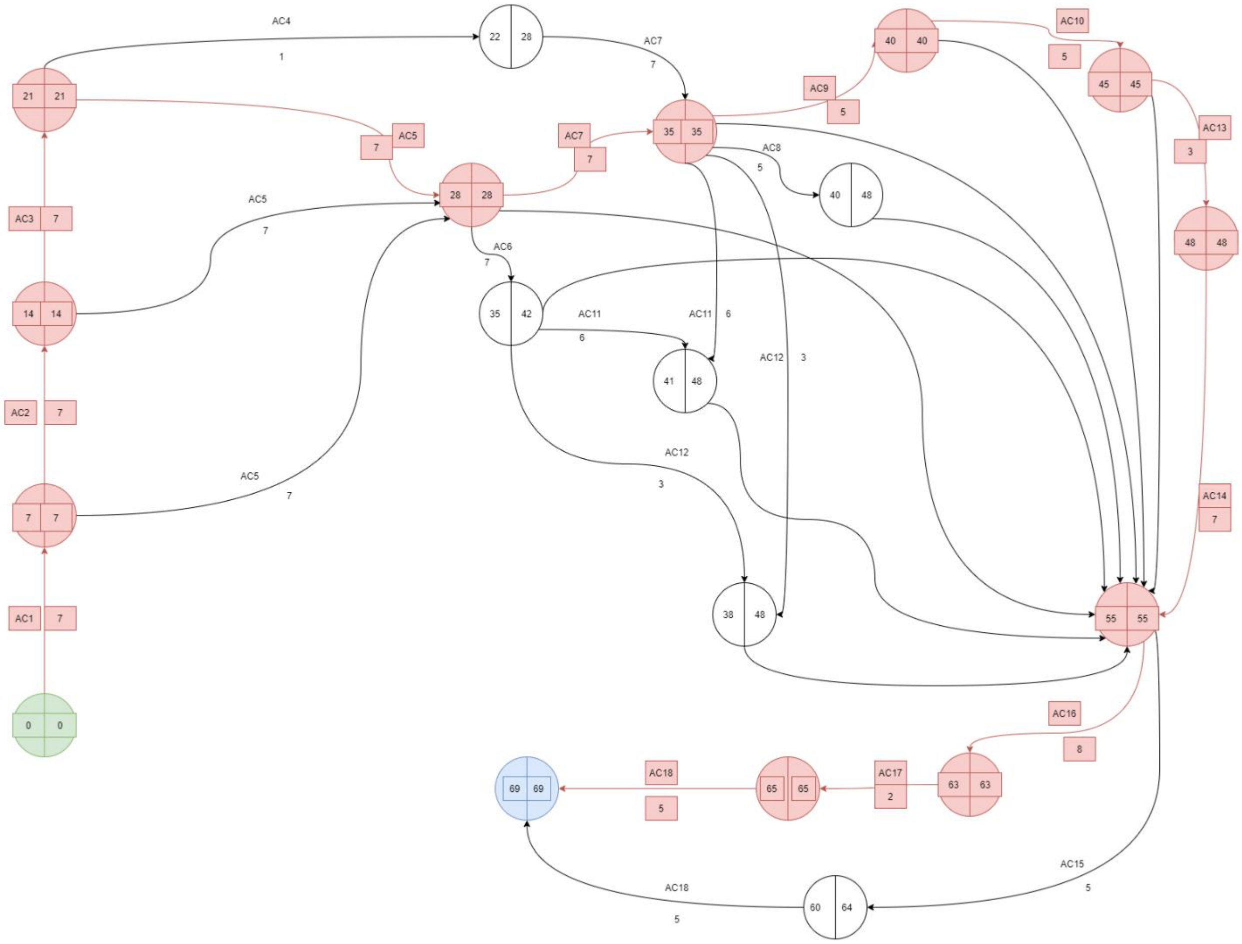
Duration: 69 days

6.1 Activity table

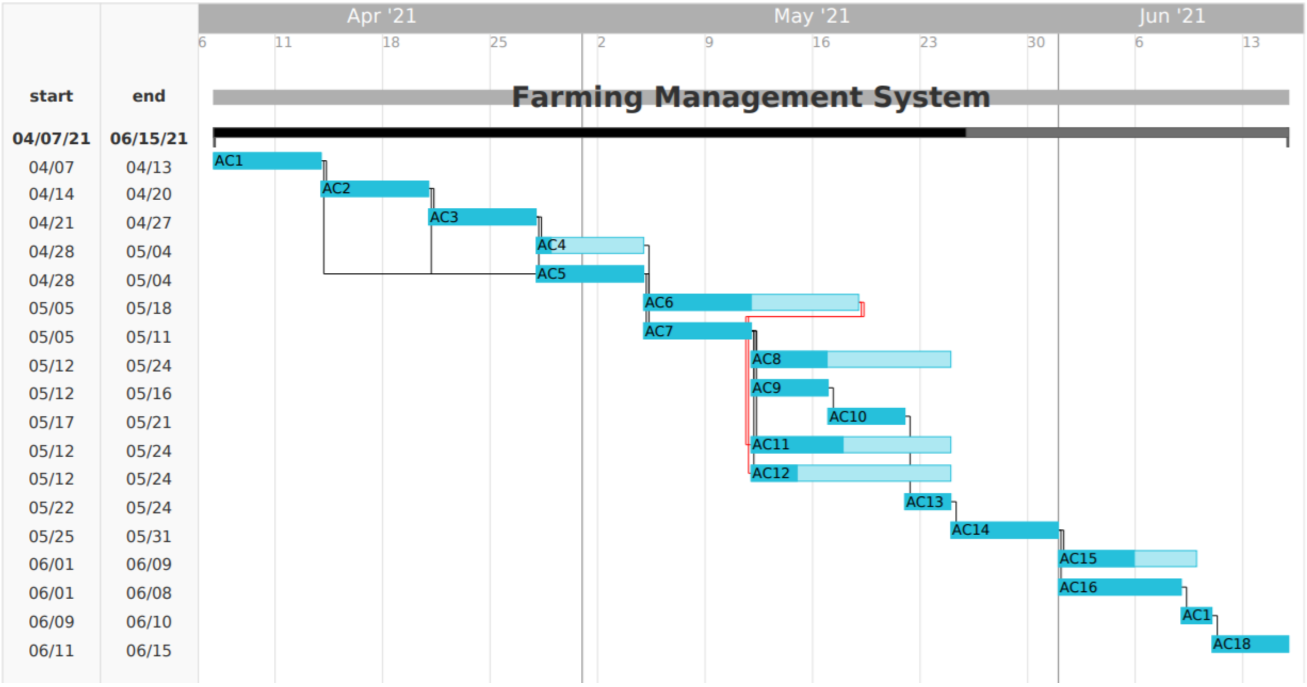
No.	Activity	Duration(days)	Dependencies	Worked
AC1	Choose topic	7		Team
AC2	Research	7	AC1	Team
AC3	Requirement Analysis	7	AC2	Team
AC4	Sketches	1	AC3	Kejdi Domi
AC5	Revise Requirements	7	AC1, AC2, AC3	Team
AC6	Use cases + User Scenarios	7	AC5	Kejdi Domi / Anila Hoxha
AC7	Use Case Diagrams	7	AC5, AC4	Kejdi Domi / Anila Hoxha
AC8	Activity Diagrams	5	AC7	Anila Hoxha
AC9	Sequence Diagrams	5	AC7	Batjona Tahiraj / Alessia Toli
AC10	Collaboration Diagrams	4	AC9	Batjona Tahiraj / Alessia Toli

AC11	Entity Relationship Diagram	6	AC6, AC7	Anila Hoxha
AC12	Data Flow Diagrams	3	AC6, AC7	Batjona Tahiraj / Alessia Toli
AC13	Class Diagram	3	AC10	Kejdi Domi
AC14	Revision of Diagrams	7	AC5, AC6, AC7, AC8, AC9, AC10, AC11, AC12, AC13	Team
AC15	Frontend Coding	5	AC14	Endri Seferi / Anila Hoxha
AC16	Backend Coding	8	AC14	Kejdi Domi / Anila Hoxha
AC17	Testing	2	AC16	Alessia Toli
AC18	Documentation	4	AC17, AC15	Team

6.2 PERT chart



6.3 Gantt chart



APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview
BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

Appendix D. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode
Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class
Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects
Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature
A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus
Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response
Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.