SOFTWARE ENGINEERING TEAM PROJECT

SSE 4301-2

TASK:

SOFTWARE DESIGN DOCUMENT

PROJECT MANAGER:

MUHAMMAD RAFIQ BIN ABD RASHID

SUPERVISED BY:

DR SALFARINA BINTI ABDULLAH

ADVISED BY:

EN. MD SHAHZAR MD. IDRIS

PREPARED BY:

NO.	STUDENT NAME	MATRIC NUMBER
1	IZZAT HAKIMI BIN IKMAL HISHAM	208335
2	NOOR SYAHIRAH BINTI MOHD SABRI	207542
3	SOFEA EZZATY BINTI MOHD KHUSRI	207084
4	DANISH AL-MUHAIMIN BIN MUSTAFFA	205647
5	JULIA NURFADHILAH BINTI MOHAMAD FAUZI	208256
6	MUHAMMAD HARITH THAQIF BIN MOHD HAIRUDDIN	205823
7	MUHAMMAD RAFIQ BIN ABD RASHID	206214

LORAFISH

Software Design Document

for

LoRa Based Alert System for Aquatank Water Quality (Condition) and Fish Measurement System (LoRaFish)

Prepared by Team LoRaFish

Faculty of Computer Science and Information Technology, UPM

Version 1.0.0

5/5/2023

GROUP MEMBERS:

STUDENT NAME	MATRIC NUMBER
IZZAT HAKIMI BIN IKMAL HISHAM	208335
NOOR SYAHIRAH BINTI MOHD SABRI	207542
SOFEA EZZATY BINTI MOHD KHUSRI	207084
DANISH AL-MUHAIMIN BIN MUSTAFFA	205647
JULIA NURFADHILAH BINTI MOHAMAD FAUZI	208256
MUHAMMAD HARITH THAQIF BIN MOHD HAIRUDDIN	205823
MUHAMMAD RAFIQ BIN ABD RASHID	206214

DOCUMENT APPROVAL

Review	Name	Date
Reviewed by: _____ (**SUPERVISOR**)	SALFARINA BINTI ABDULLAH	
Approved by:  **(Client)**	EN. MD SHAHZAR MD. IDRIS	5/5/2023

TABLE OF CONTENTS

1. Introduction

The project is to notify and alert regarding changes on water quality and measure length of fish in each aquatank. The system should be able to display the status of water quality alongside length of fish, allow management of marine life, water quality, and alarm for each aquatank and analyse the cost of maintaining each aquatank.

This design document presents the designs used or intended to be used in implementing the project. The designs described, follow the requirements specified in the Software Requirements Specifications document prepared for the project.

1.1. Purpose

The purpose of this Software Design Document (SDD) is to provide a detailed and comprehensive description of the software design for the LoRa Based Alert System for Aquatank Water Quality (Condition) and Fish Measurement System (LoRaFish). This document serves as a guide for the development team, providing a clear understanding of the system's architecture, modules, components, and interfaces. Additionally, the SDD serves as a reference for future software maintenance and enhancements. The document is intended for the development team, project managers, and other stakeholders involved in the software development process.

1.2. Scope

This document gives a detailed description of the software architecture of the Aquatank Water Quality (Condition) and Fish Measurement System (LoRaFish) system. It specifies the structure and design of some of the modules discussed in the SRS. It also displays some of the use cases that had transformed into sequence and activity diagrams. The class diagrams show how the programming team would implement the specific module.

1.3. References

The user of this SDD may need the following documents for reference:

- IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

- Team LoRaFish, 2010. Software Requirements Specification for Aquatank Water Quality (Condition) and Fish Measurement System (LoRaFish). Last modified: Apr. 8, 2023.

1.4. Overview

The standards for software design documentation outlined in "IEEE Recommended Practices for Software Design Documentation" were followed in the writing of this work.
Sections 3 – 5 contain discussions of the designs for the project with diagrams, section 6 shows samples of UI from the system.

2. Design considerations

2.1. Assumptions

The following are the main assumptions;

- The users have basic computer literacy and are comfortable with web-based applications.
- The sensors used in the aqua tanks are in working condition and accurately measure the parameters required for monitoring.
- The internet connectivity at the location of the aqua tanks is stable and reliable.

2.2. Constraints

The following are constraints in building the system are;

- This project must be developed within a specified budget and timeframe.
- The accuracy of the sensors depends on the type and quality of the sensors used.
- The system must have reliable connectivity to ensure that data can be transmitted from the sensors to the monitoring system.

2.3. System environment

The web based system is designed to work on all operating systems. The system is accessible through any laptop and desktop that is connected to the server. It is accessible at all times.

2.4. Design methodology

The system has been created with adaptability in mind to enable future development and modifications. The system has been separated into controllable procedures that are organised into sub-modules and modules that have been constructed using abstraction.

3. Architecture

3.1. System design

The system design of this project aims to develop a web-based application that enables PJBUMIHES to monitor and analyse data from their aqua tanks, while allowing for easy management of sensors, aqua tank marine life, aqua tank acidity, alarms, and resources through a user-friendly dashboard.

3.1.1 Block diagram

The block diagram below shows the principal parts of the system and their interactions.

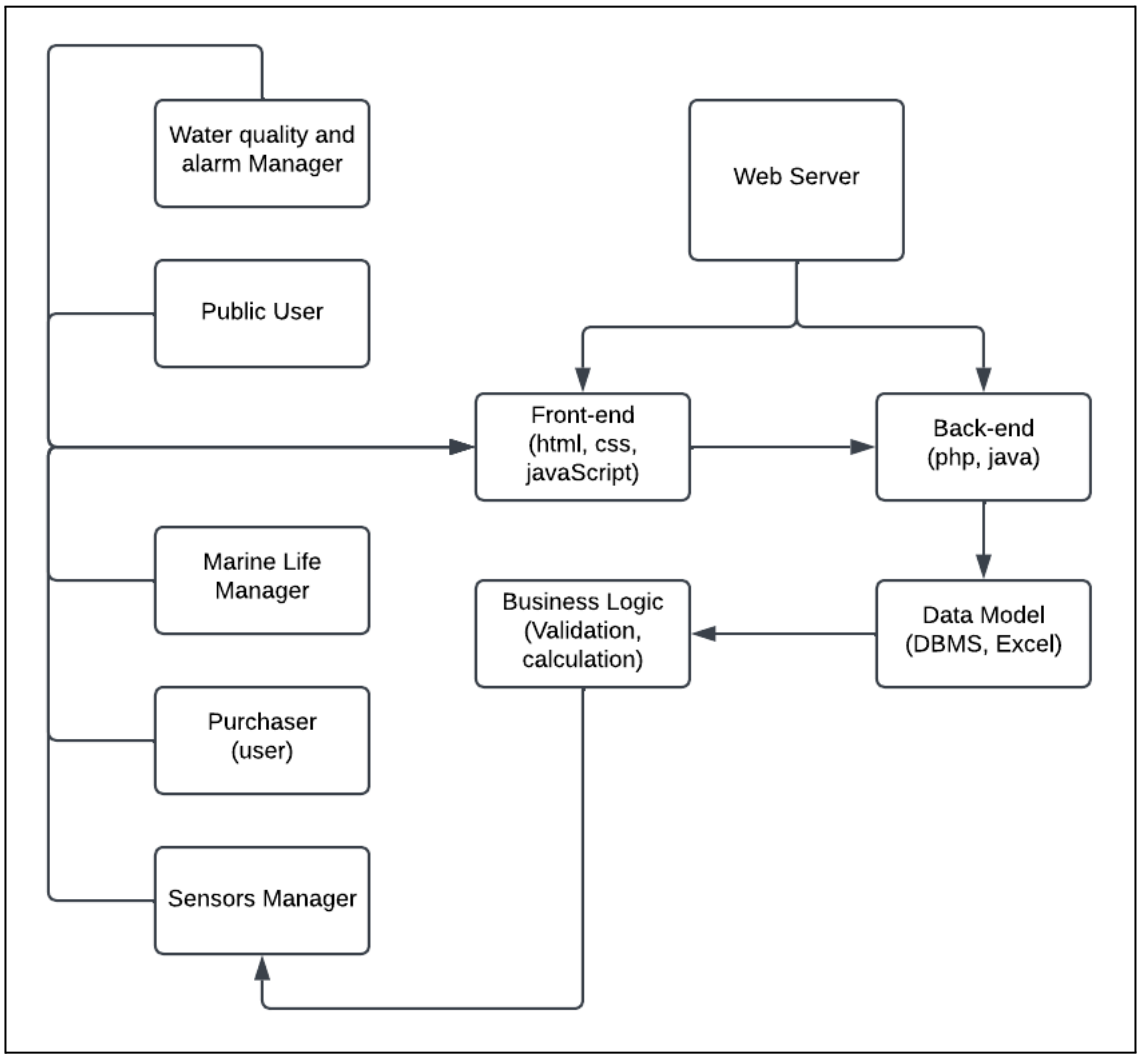


Figure 1. Block diagram for LoraFish

3.1.2 Context diagram

The context diagram provides an overview of the system's external entities and how they interact with the system, including users and other systems. It serves as a high-level representation of the system's boundaries and its relationships with the outside world.

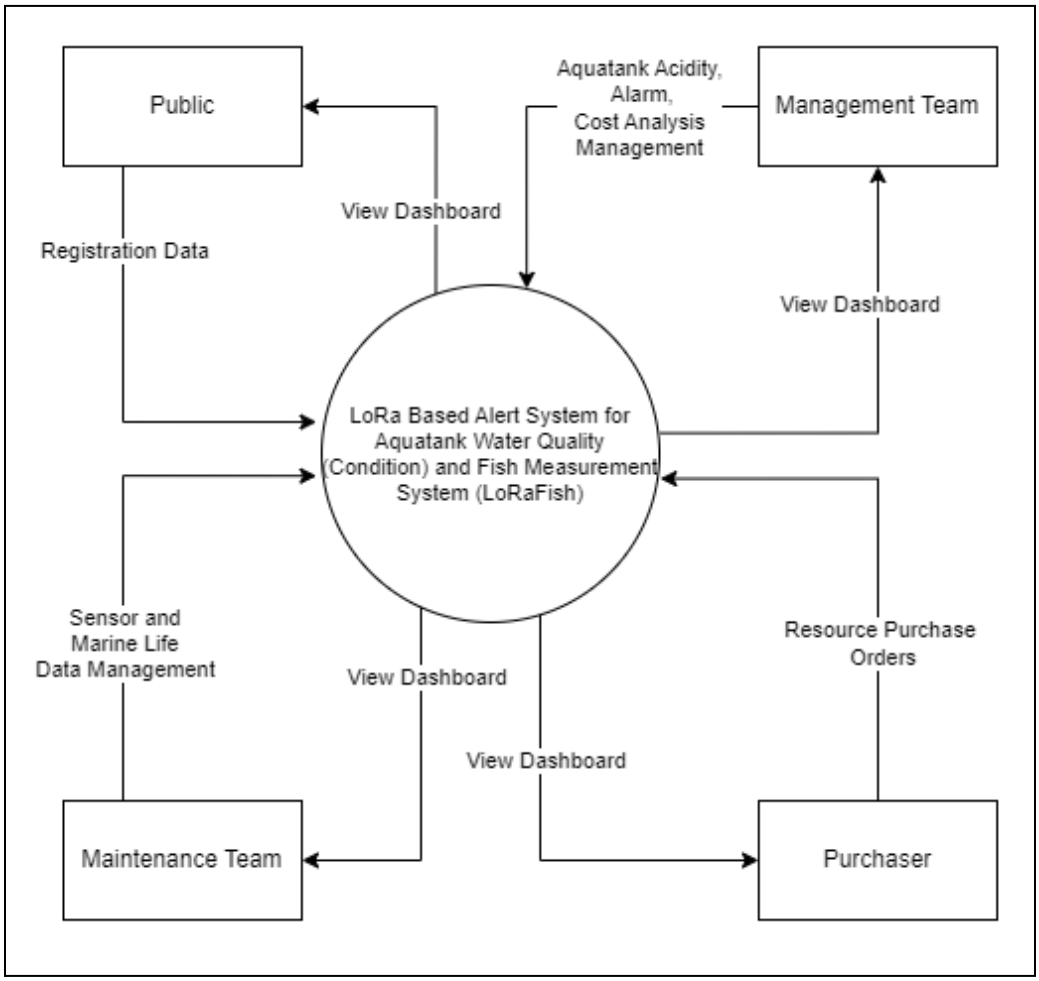


Figure 3.2 *LoRa Based Alert System for Aquatank Water Quality (Condition) and Fish Measurement System (LoRaFish)*

4. Data design

4.1. Data description

The data design section of this document outlines the description of the data storage and retrieval system for the web-based aqua tank monitoring application. The system is designed to use mySQL database management system and Apache web server to integrate with the application.

4.2. Data dictionary

The data dictionary will provide a comprehensive description of the data tables used in each use case of the system. This document will serve as a reference guide for all data elements, their definitions, and relationships within the system.

4.2.1 User Table

Field	Type	Null	Description
Email	varchar(20)	No	A unique identifier assigned to each user to access the system.
Username	varchar(15)	No	A unique identifier assigned to each user to access the system. Easier to remember.
Password	varchar(20)	No	Provide a secure way to protect user accounts from unauthorised access.

Table 4.2.1 User Table

4.2.2 Manage Sensor Table

Field	Type	Null	Description
sensorID	varchar(10)	No	A unique identifier assigned to each aqua tank sensor in the system.
dateAdded	Date	No	The date the sensor is added to the system.
status	Binary	No	The status of the sensor, whether on active or inactive.

Field	Type	Null	Description
tankNo	varchar(1)	No	The tank number which the sensor is set to.
sensorType	varchar(20)	No	The type of the sensor.
sensorInput	varchar(20)	No	The input of the sensor.
inputUnit	varchar(10)	No	The unit of input of the sensor.
lastServiced	Date	No	The last date that the sensor is serviced.

Table 4.2.2 Manage Sensor Table

4.2.3 Aquatank Marine Life Table

Field	Type	Null	Description
tankNo	varchar(50)	No	A unique identifier assigned to each aqua tank in the system.
fishType	varchar(50)	No	The species or variety of fish.
avgfishLength	double	No	The average length of the fish is millimetres.
avgfishWeight	double	No	The average weight of an individual fish in grams.
fishQuantity	double	No	The number of fish in a specific tank.
totalFishWeight	double	No	The combined weight of a fish type in a tank.
lastdateAdded	Date	No	The date when the fish arrived in the tank.

Table 4.2.3 Aquatank Marine Life Table

4.2.3.1 Marine Life Table

Field	Type	Null	Description
fish_ID	int(11)	No (auto increment)	A unique identifier for each fish in the system.
fishLength	double	No	The length of the fish in millimetres.

Field	Type	Null	Description
fishWeight	double	No	The weight of an individual fish in grams.
fishGrowthRate	double	No	The rate at which fish are growing in length per day.
dateAdded	Date	No	The date when the fish arrived in the tank.

Table 4.2.3.1 Marine Life Table

4.2.4 Water Condition Table

Field	Type	Null	Description
Date	Date	No	The date when the data from sensor is retrieved.
Time	Time	No	The time when the data from sensor is retrieved.
TankNo	varchar(50)	No	A unique identifier assigned to each aquatank in the system.
AcidityLevel	double	No	The acidity level of water in the aquatank in pH.
OxygenLevel	double	No	The oxygen level of water in the aquatank in mg/L.
HydrogenLevel	double	No	The hydrogen level of water in the aquatank in mg/L.
NitrateLevel	double	No	The nitrate level of water in the aquatank in mg/L.
CarbonDioxideLevel	double	No	The carbon dioxide level of water in the aquatank in mg/L.
MercuryLevel	double	No	The mercury level of water in the aquatank in mg/L.
Temperature	Int (11)	No	The temperature of water in the aquatank in degree celsius.

Table 4.2.4.3 Water Condition Table

4.2.5 Manage Alarm Table

Field	Type	Null	Description
TankNo	varchar(50)	No	A unique identifier assigned to each aquatank in the system.
MaxAcidityLevel	double	No	The maximum value of good acidity level of water in the aquatank in pH.
MinAcidityLevel	double	No	The minimum value of good acidity level of water in the aquatank in pH.
MaxOxygenLevel	double	No	The maximum value of good oxygen level of water in the aquatank in mg/L.
MinOxygenLevel	double	No	The minimum value of good oxygen level of water in the aquatank in mg/L.
MaxHydrogenLevel	double	No	The maximum value of good hydrogen level of water in the aquatank in mg/L.
MinHydrogenLevel	double	No	The minimum value of good hydrogen level of water in the aquatank in mg/L.
MaxNitrateLevel	double	No	The maximum value of good nitrate level of water in the aquatank in mg/L.
MinNitrateLevel	double	No	The minimum value of good nitrate level of water in the aquatank in mg/L.
MaxCarbonDioxide Level	double	No	The maximum value of good carbon dioxide level of water in the aquatank in mg/L.
MinCarbonDioxide Level	double	No	The minimum value of good carbon dioxide level of water in the aquatank in mg/L.
MaxMercuryLevel	double	No	The maximum value of good mercury level of water in the aquatank in mg/L.
MinMercuryLevel	double	No	The minimum value of good mercury level of water in the aquatank in mg/L.
MaxTemperature	Int (11)	No	The maximum value of good temperature of water in the aquatank in degree celsius.

Field	Type	Null	Description
MinTemperature	Int (11)	No	The minimum value of good temperature of water in the aquatank in degree celsius.

Table 4.2.4.4 Manage Alarm Table

4.2.6 Analyze Cost Table

Field	Type	Null	Description
ItemID	varchar(50)	No (auto increment)	A foreign identifier for each item that needed to be purchased.
UnitPrice	double	No	The cost of the item in Malaysian Ringgit.
Quantity	int	No	The quantity of the item.
TankNo	varchar(50)	No	A unique identifier assigned to each aqua tank in the system.
SupplierID	int(10)	No (auto increment)	A foreign identifier for each supplier company

Table 4.2.4.5 Analyze Cost Table

4.2.7 Supply Resources Table

Field	Type	Null	Description
SupplierID	int(10)	No (auto increment)	A unique identifier for each supplier company
SupplierName	varchar(50)	No	The company’s name.
ItemID	int(10)	No	A unique identifier for each item resource
ItemName	varchar(50)	No	The item description
UnitPrice	double(2)	No	Unit price for each item
DateSupply	date	No	The date recorded where resources were being purchased by the purchaser.

Field	Type	Null	Description
DateReceived	date	No	The date recorded where resources received

Table 4.2.4.6 Purchased Resource

5. Component design

"Users with appropriate permission" in the diagram refers to the users who are given exemptions or/and users of a particular level. Precise permissions are listed in Section 2.3 of SRS. For diagrams with multiple functions, the design is the same for those functions, except for parameters/methods/class.

5.1. Register

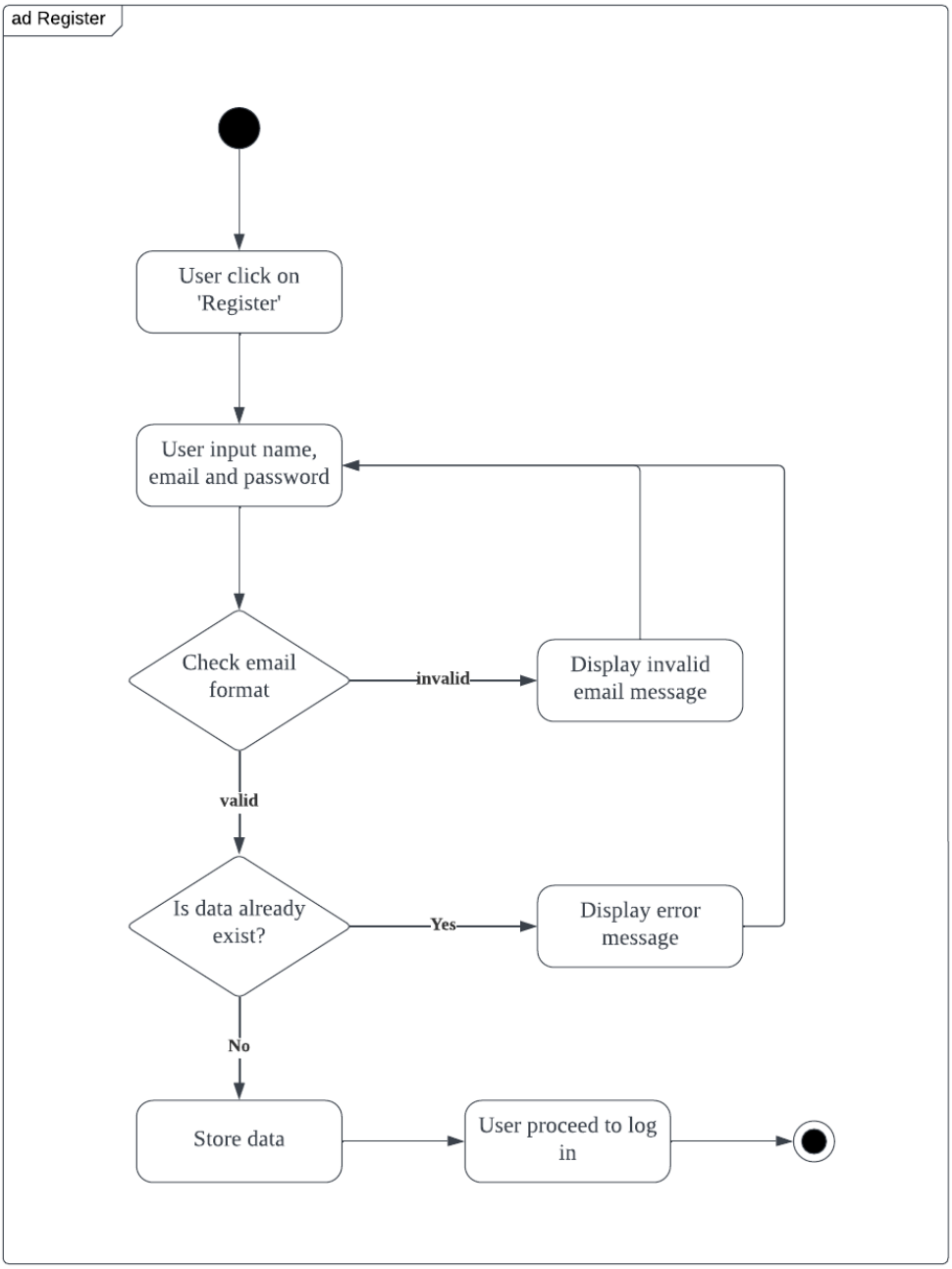


Figure 5.1.1 Activity Diagram of Register

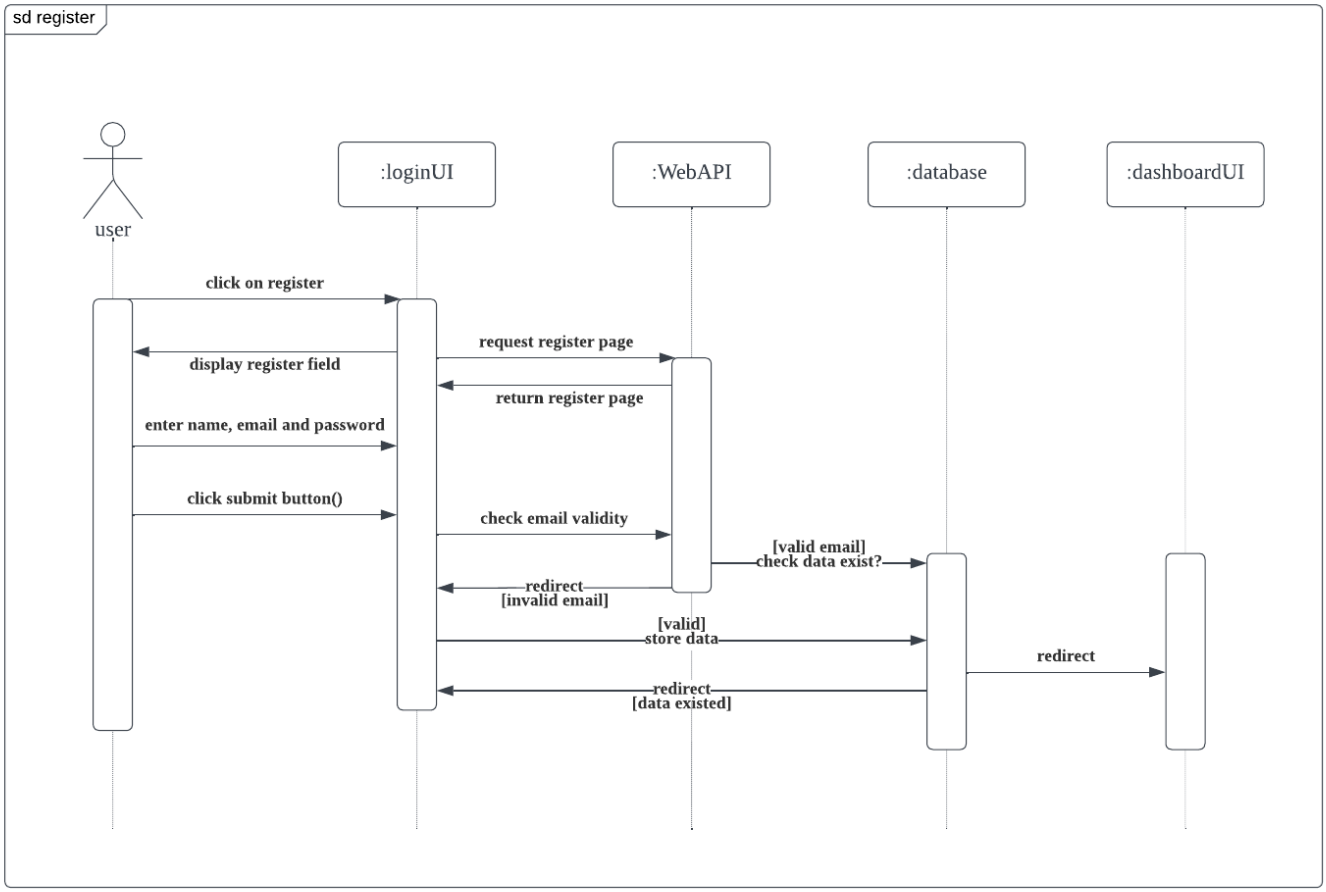


Figure 5.1.2 Sequence Diagram of Register

5.2. Login

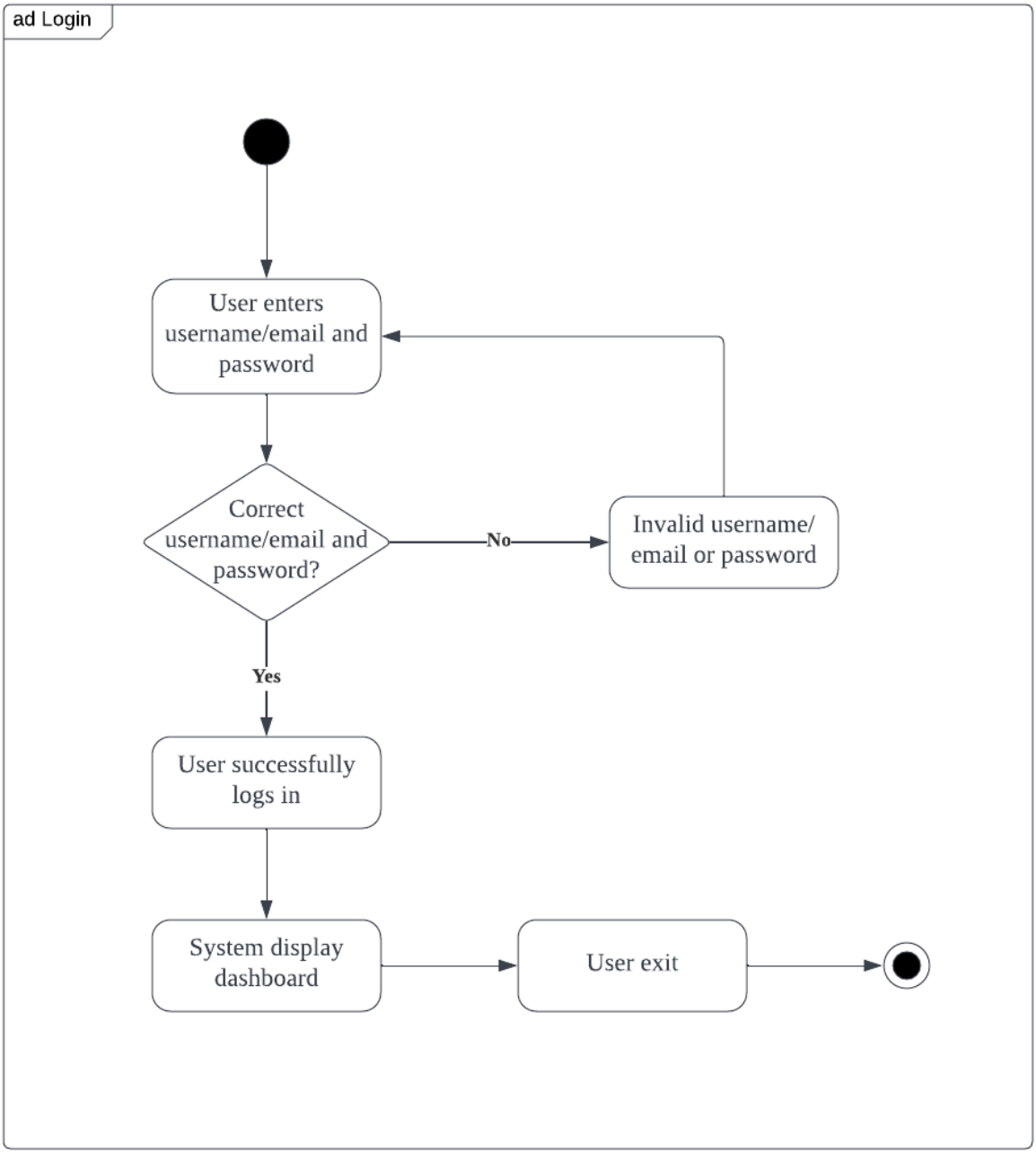


Figure 5.2.1 Activity Diagram of Login

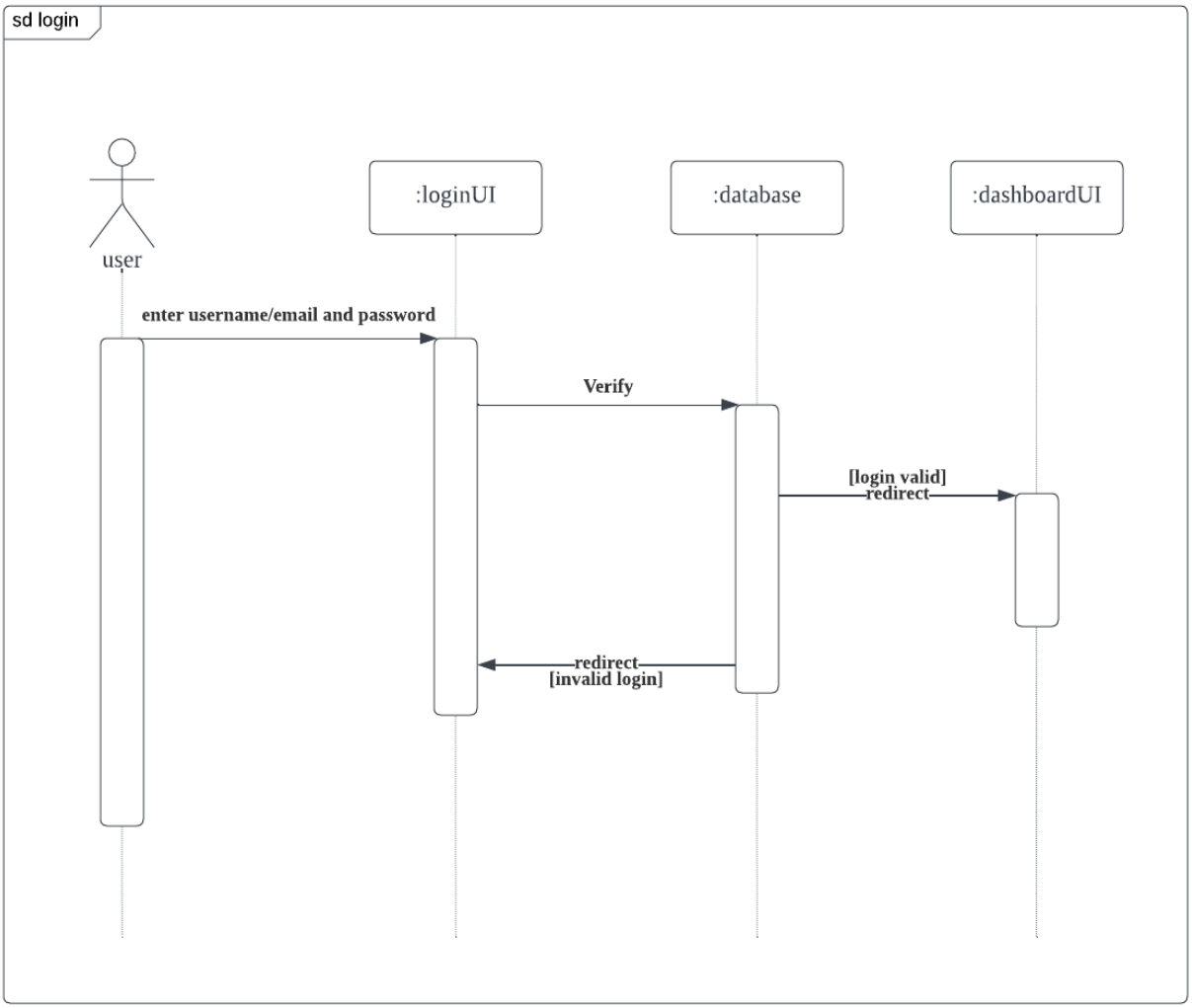


Figure 5.2.2 Sequence Diagram of Log in

5.3. View Dashboard

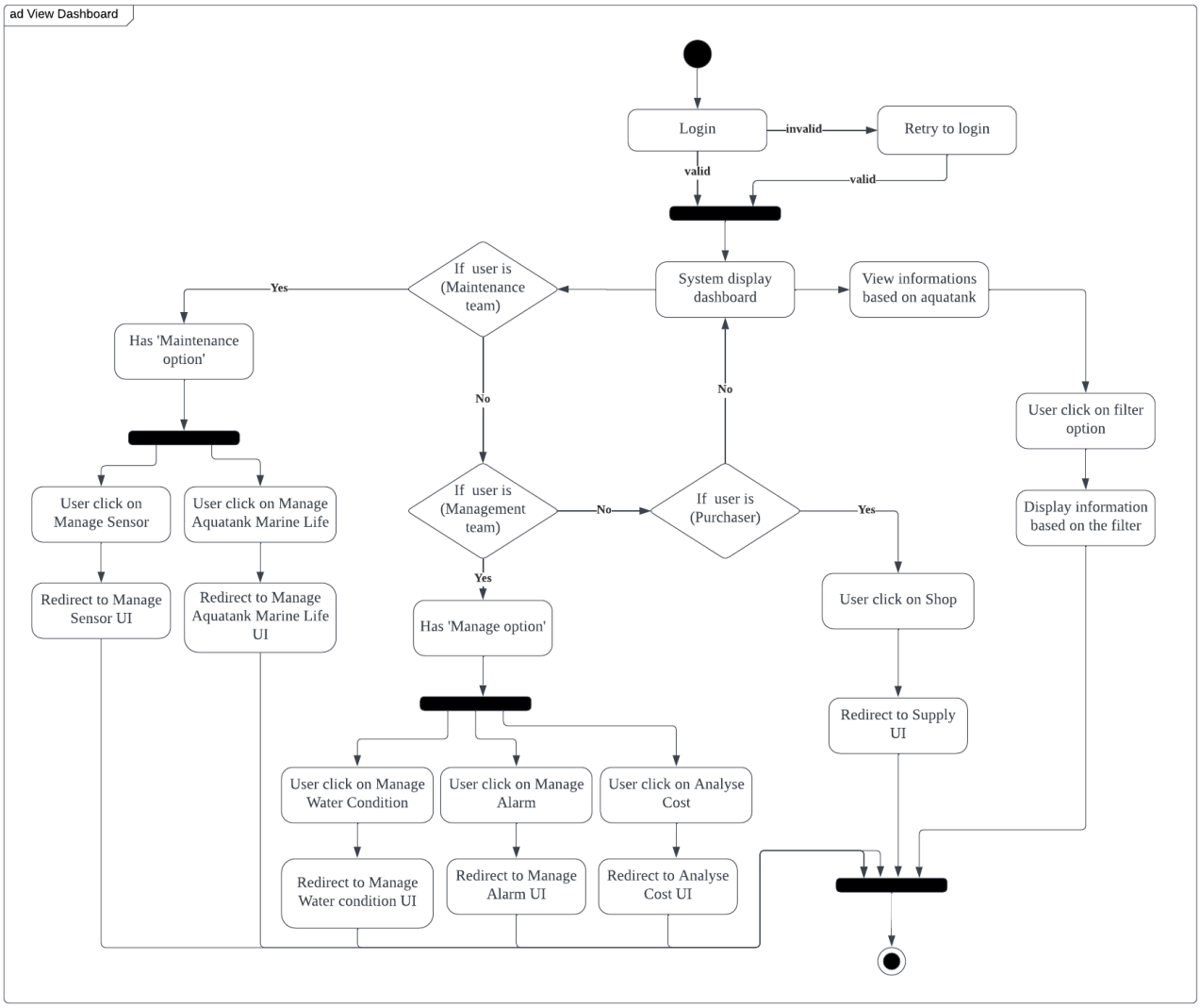


Figure 5.3.1 Activity Diagram of View Dashboard

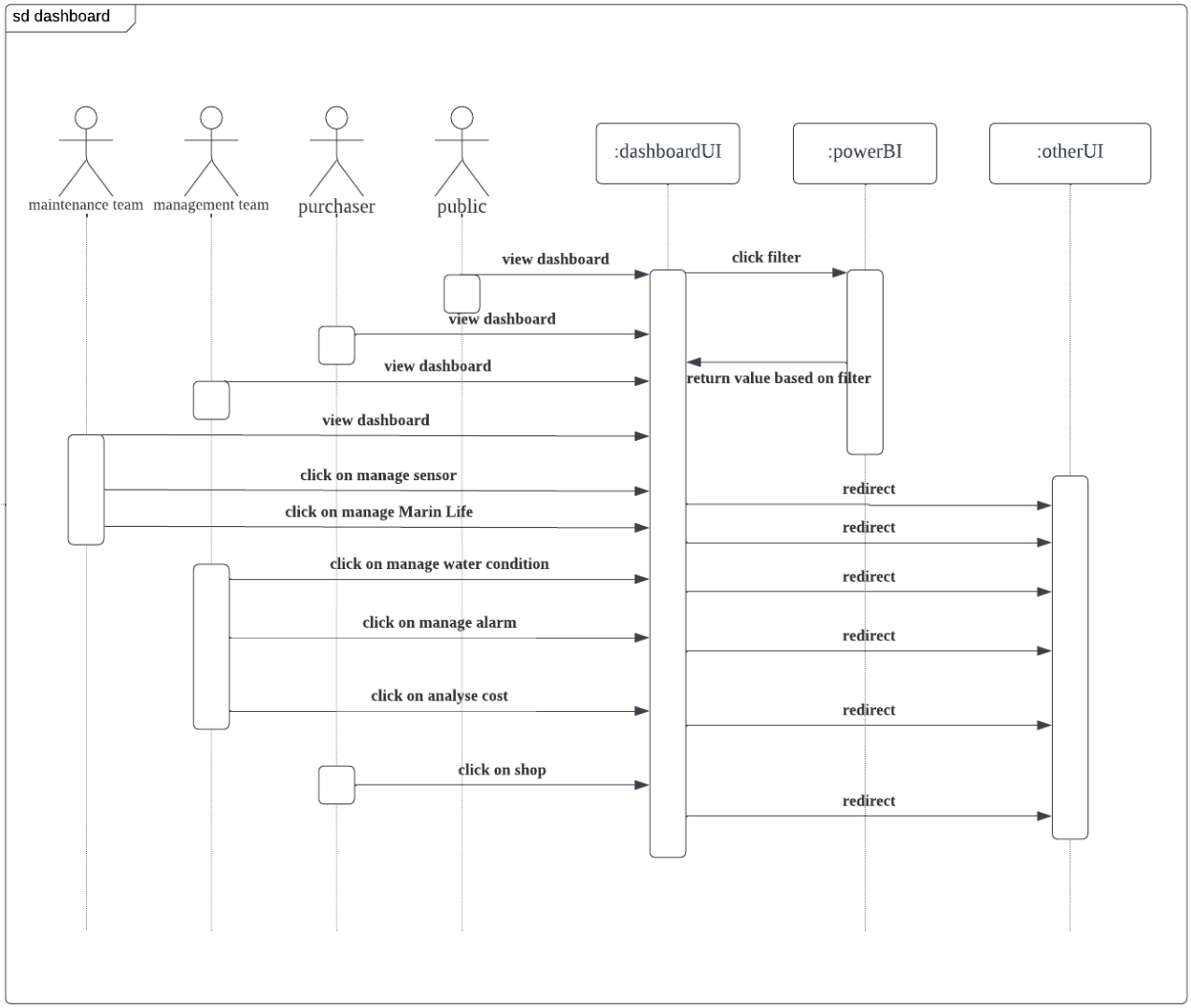
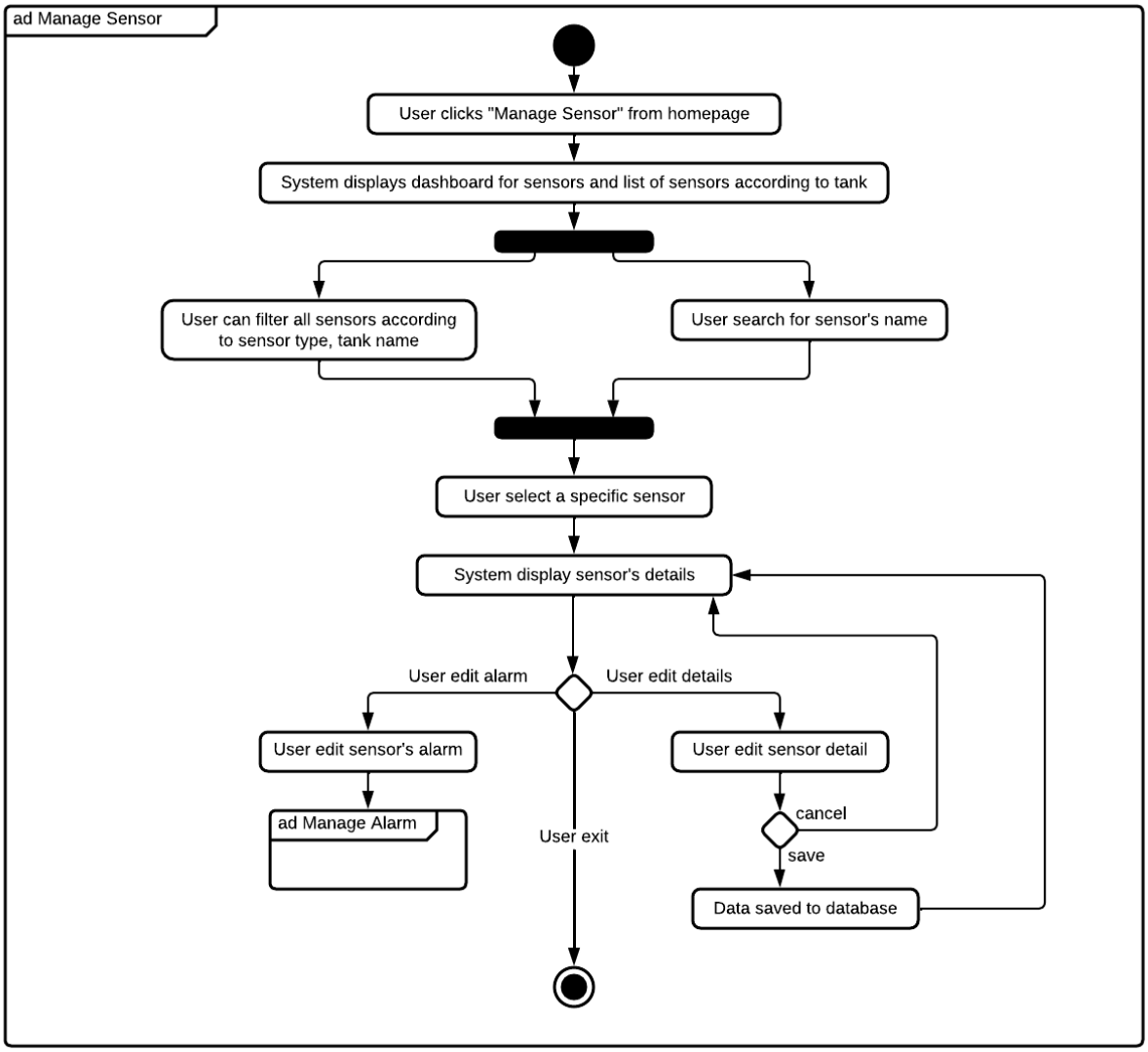
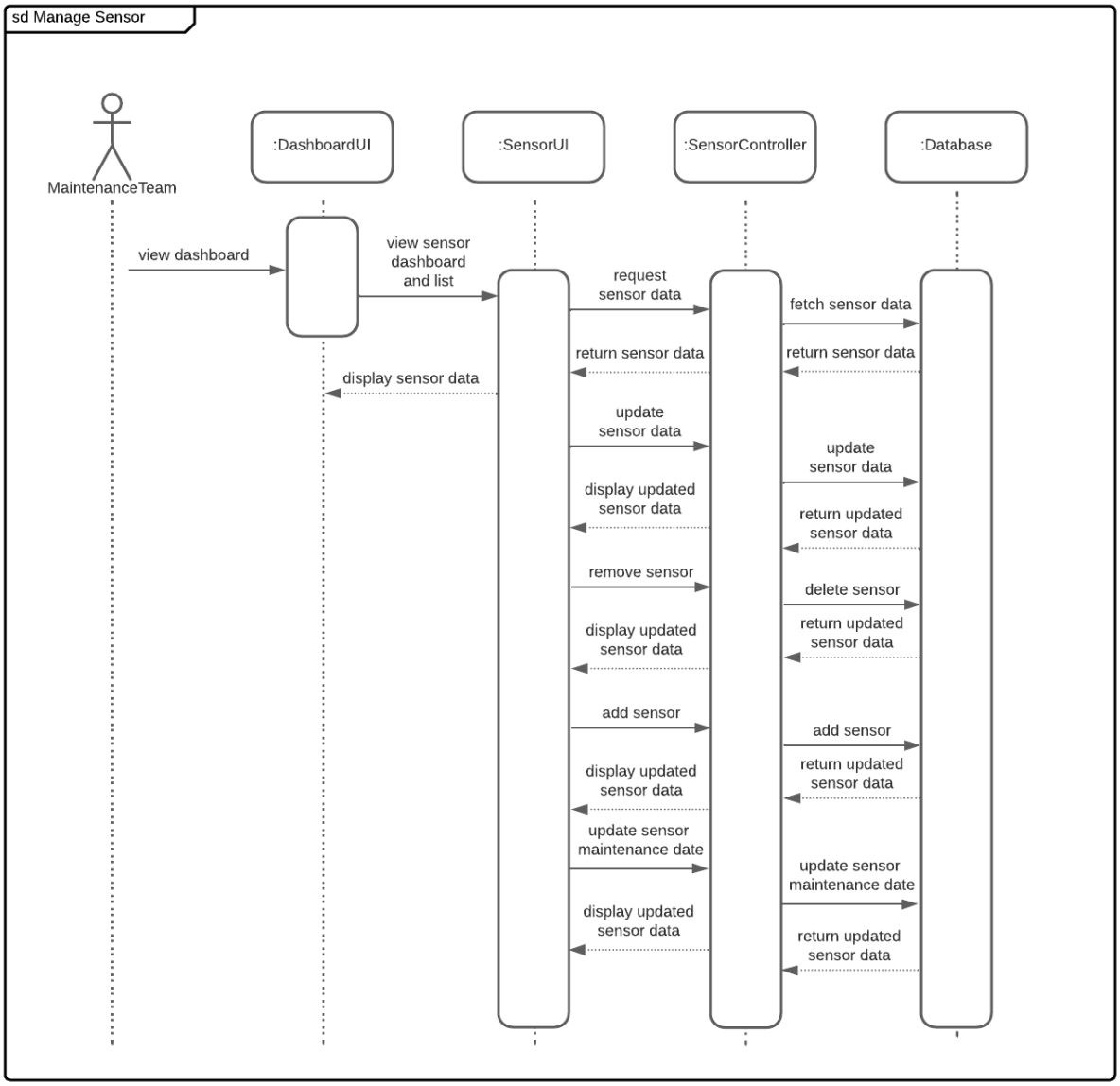


Figure 5.3.2 Sequence Diagram of View Dashboard

5.4. Manage Sensor




sd Manage Sensor
MaintenanceTeam
:DashboardUI
:SensorUI
:SensorController
:Database
view dashboard
view sensor dashboard and list
request sensor data
fetch sensor data
return sensor data
return sensor data
display sensor data
update sensor data
update sensor data
display updated sensor data
return updated sensor data
remove sensor
delete sensor
display updated sensor data
return updated sensor data
add sensor
add sensor
display updated sensor data
return updated sensor data
update sensor maintenance date
update sensor maintenance date
display updated sensor data
return updated sensor data

5.5. Manage Aquatank Marine Life

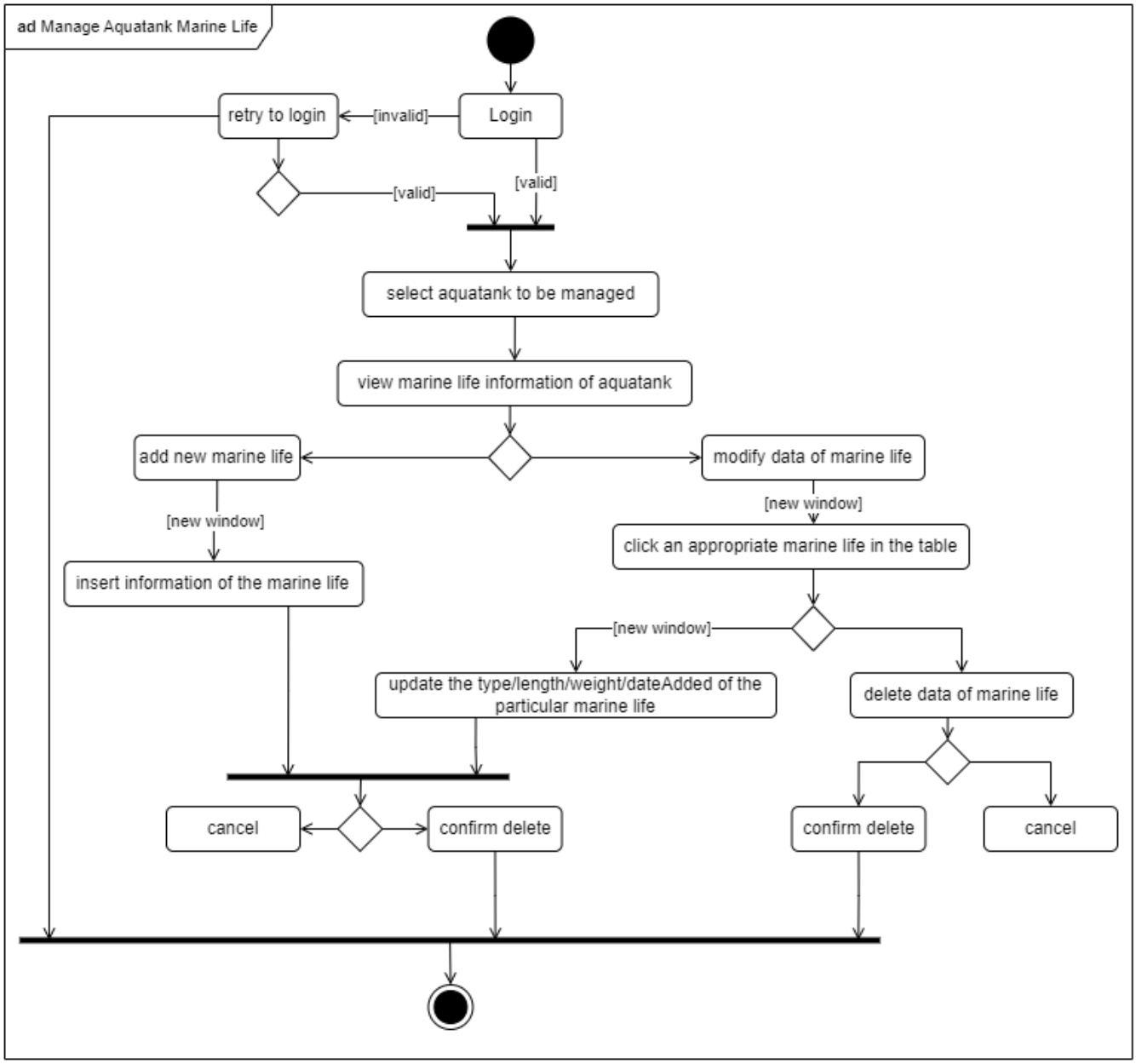


Figure 5.5.1 Activity Diagram of Manage Aquatank Marine Life

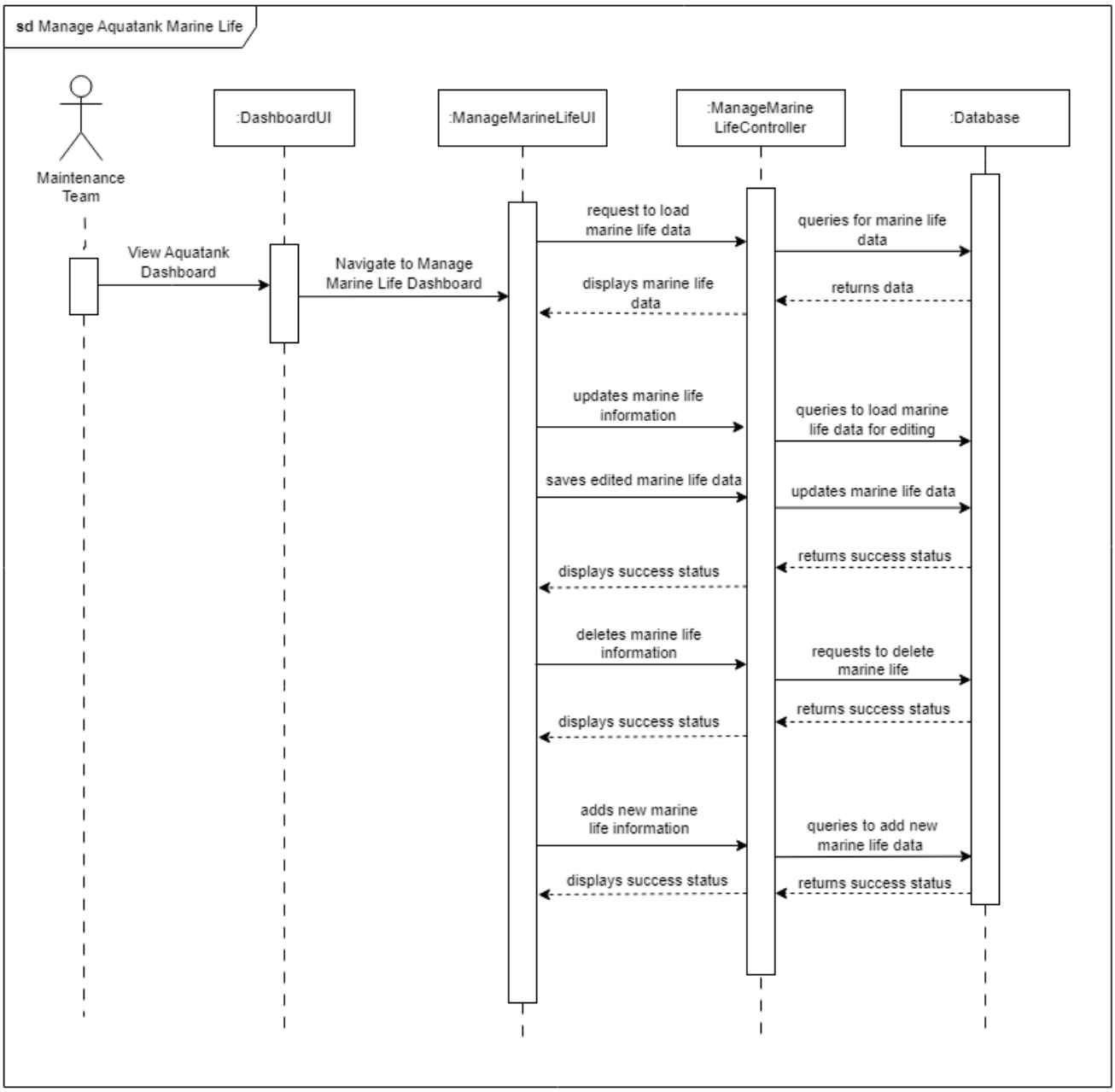


Figure 5.5.2 Sequence Diagram of Manage Aquatank Marine Life

5.6. Manage Aquatank Water Condition

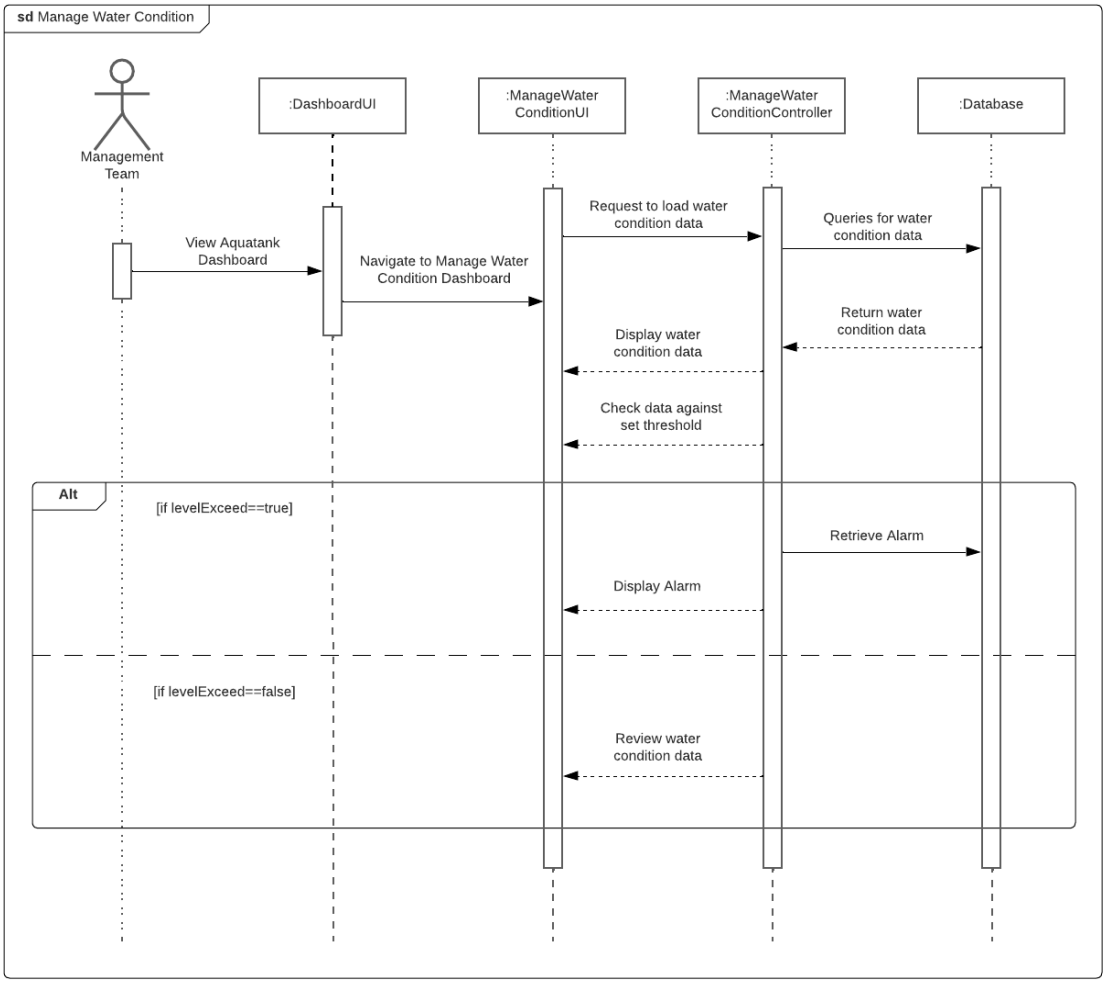


Figure 5.6.1 Sequence Diagram of Manage Aquatank Water Condition

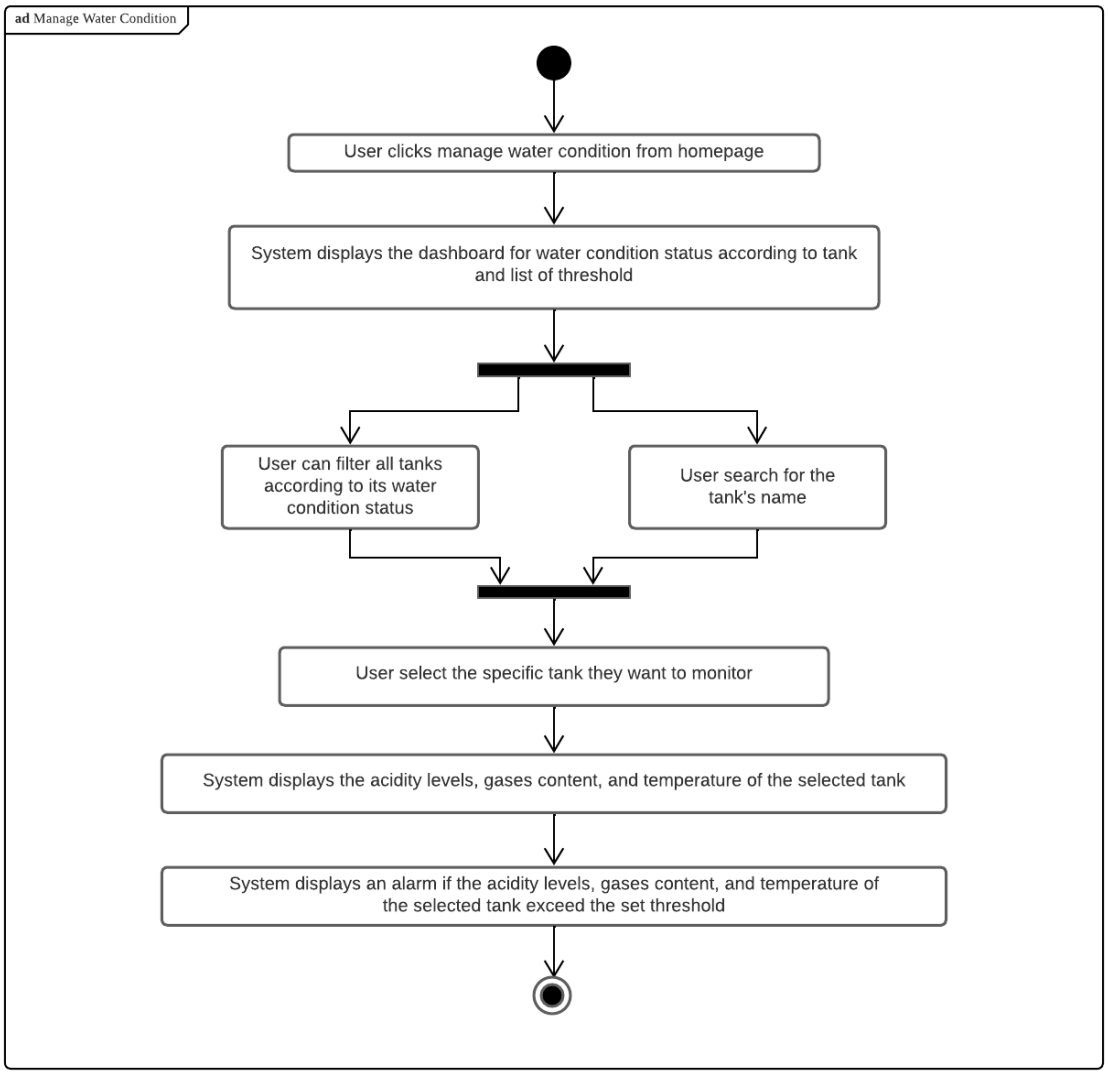
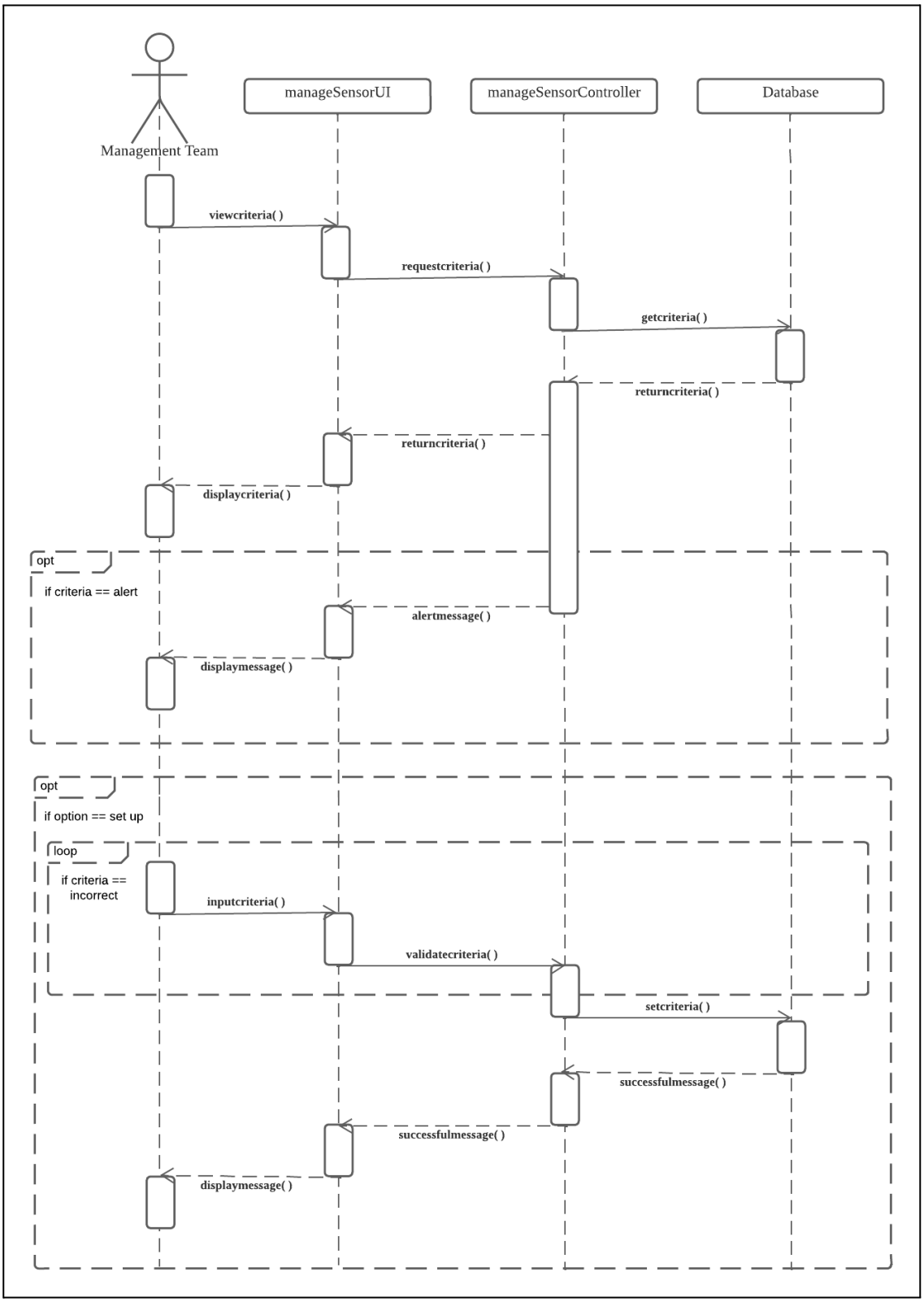
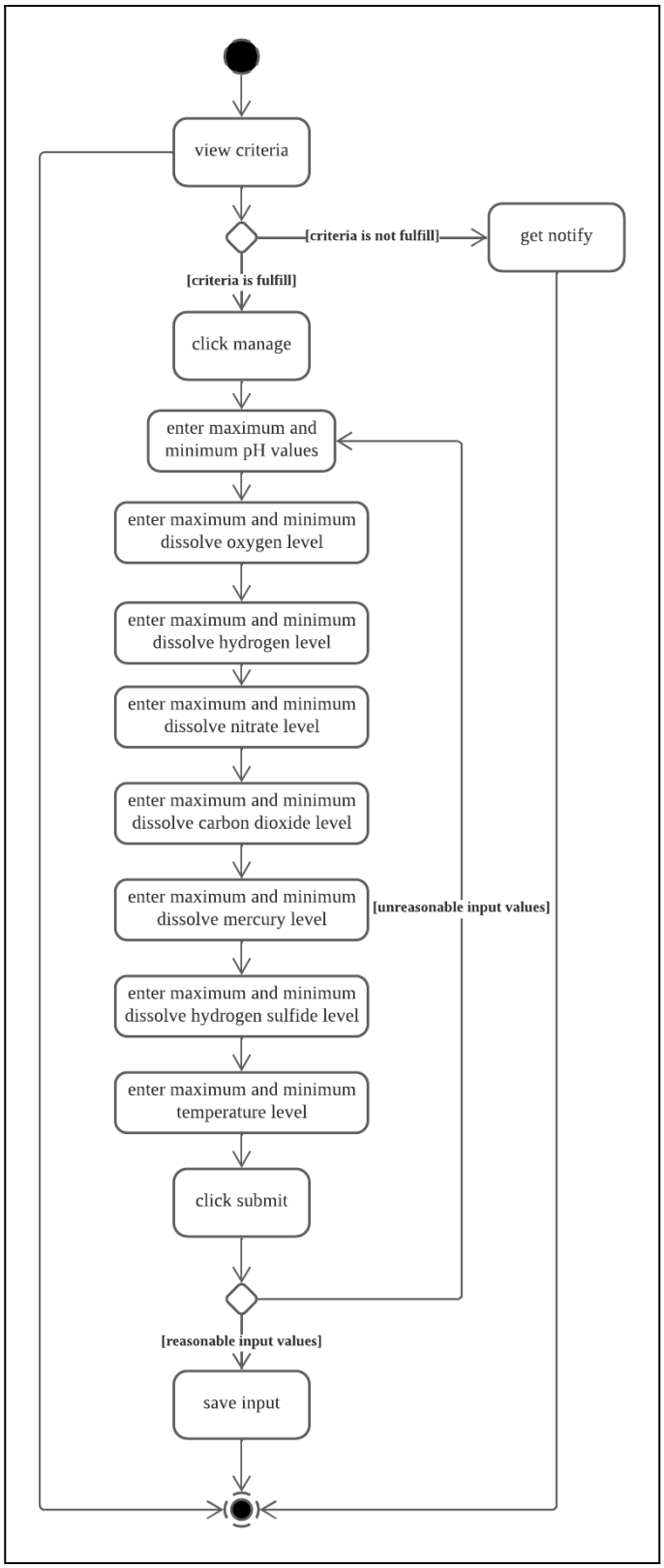


Figure 5.6.2 Activity Diagram of Manage Aquatank Water Condition

5.7 Manage Alarm





5.8. Analyse Cost

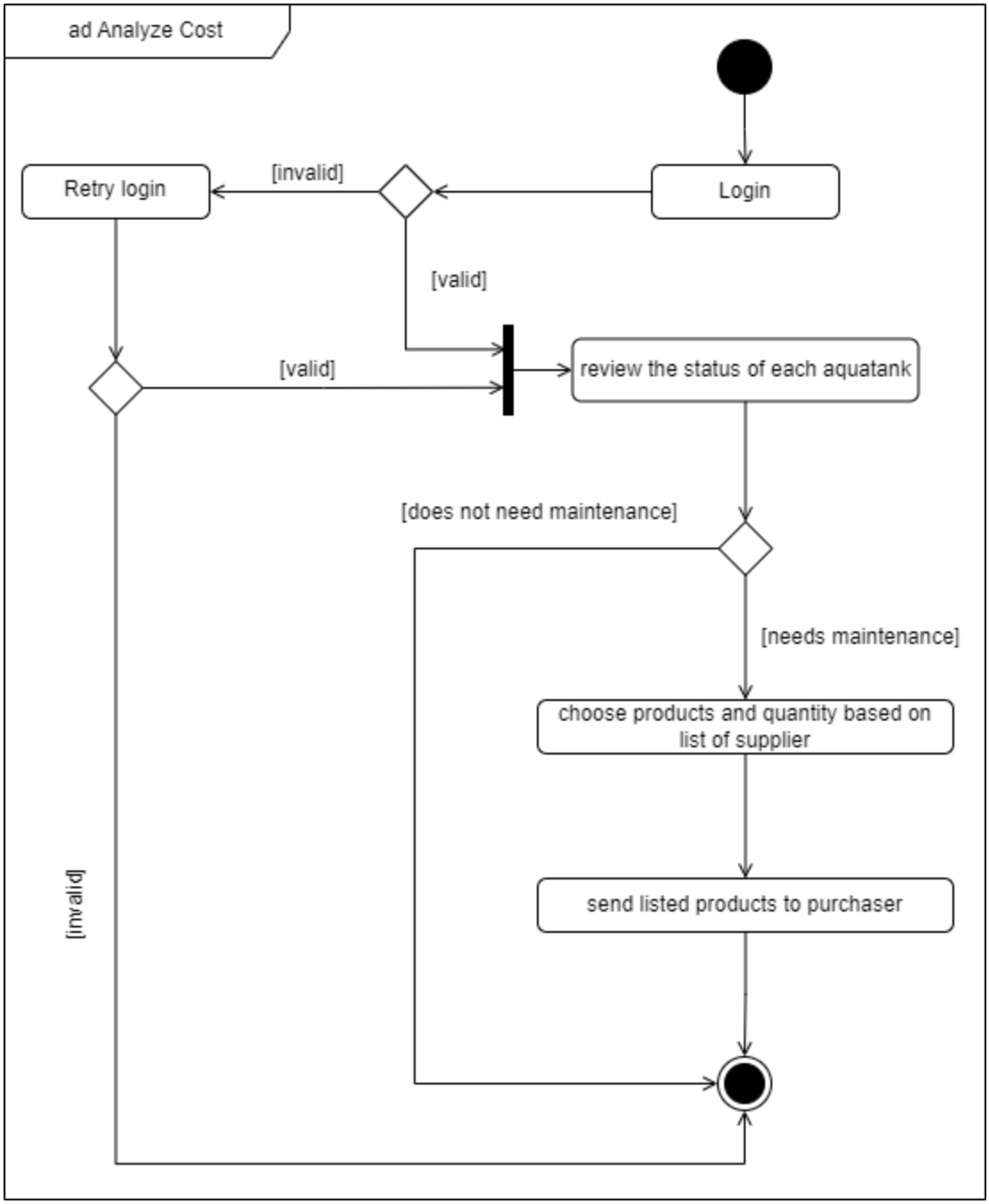


Figure 5.8.1 Activity diagram for analyse cost

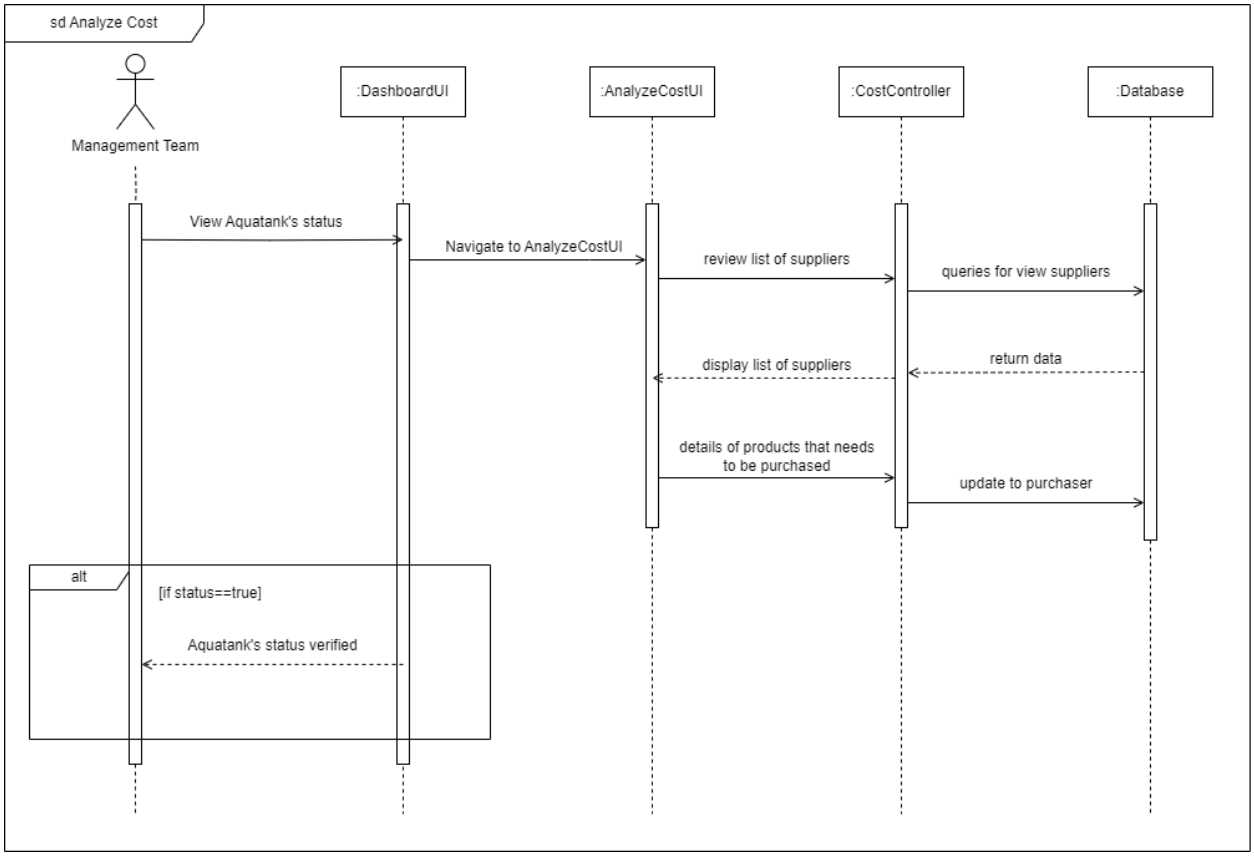


Figure 5.8.2 Sequence diagram for analyse cost

5.9 Supply Resources

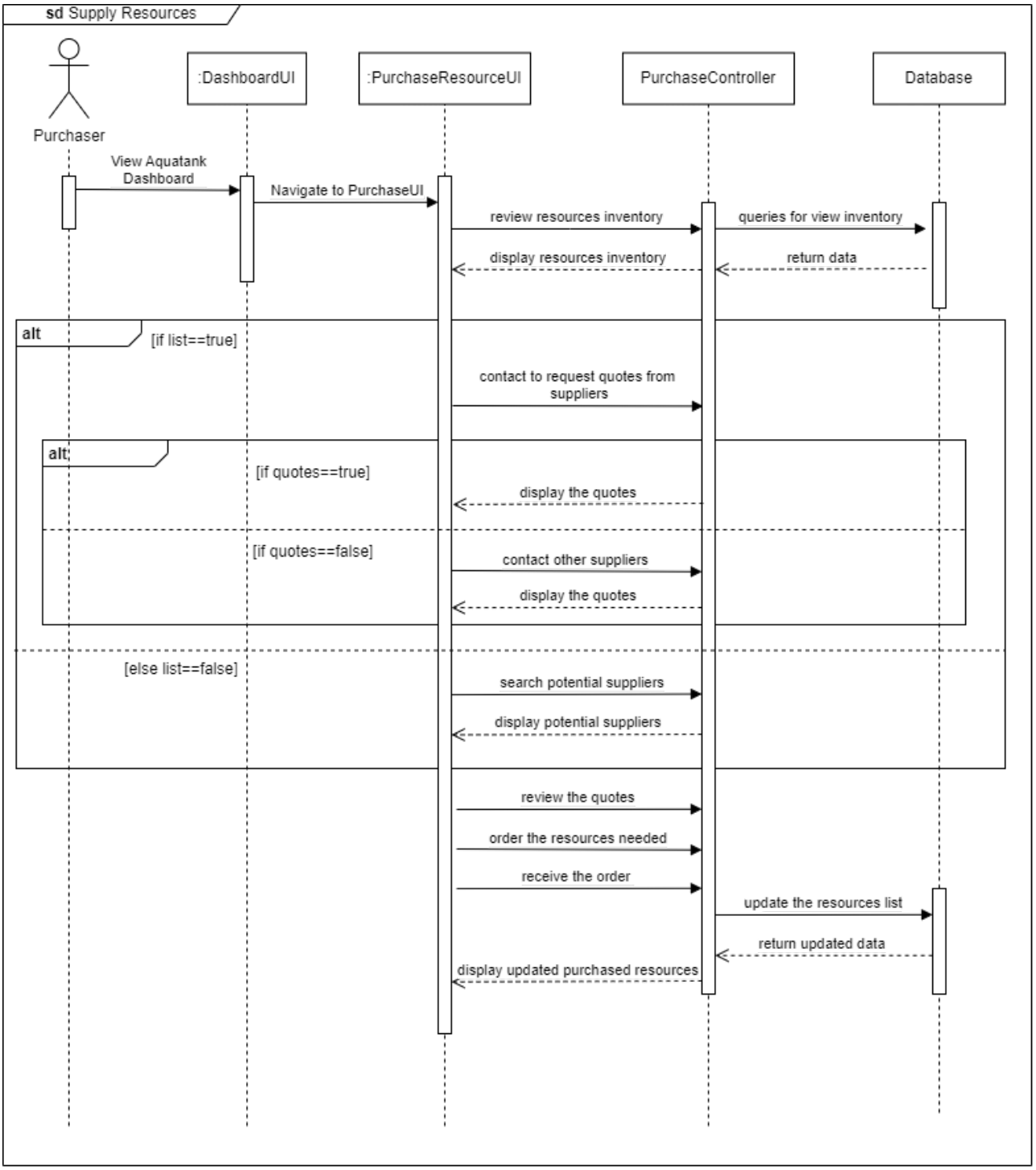


Figure 5.9a: Sequence Diagram for Supply Resources

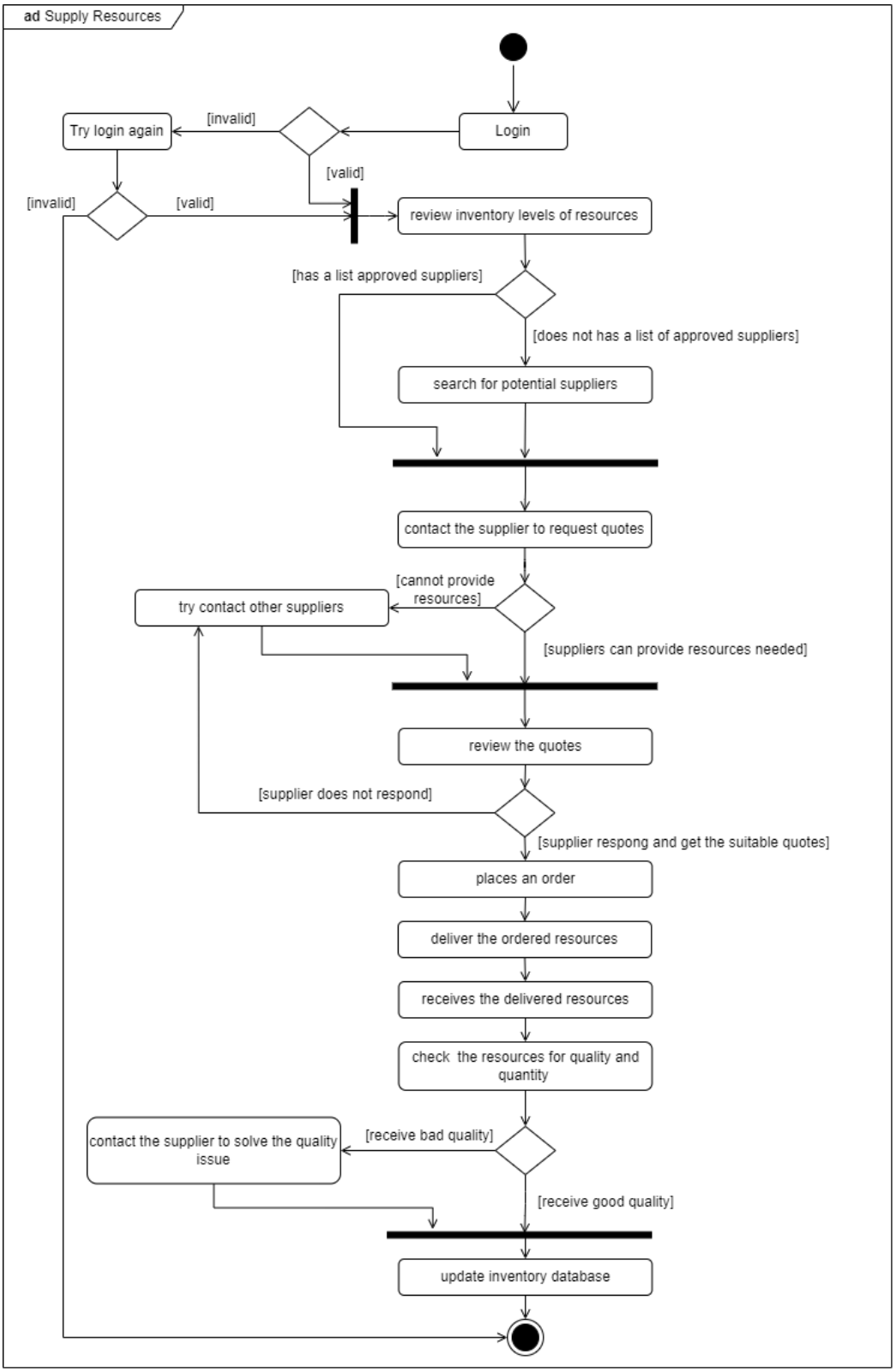


Figure 5.9b: Activity Diagram for Supply Resources

6. Software interface design

6.1. User interface design

UI is designed according to UI design principles:

- The structure principle where related items are grouped together and unrelated items are separated in the user interface (UI).
- The simplicity principle refers to using the given interface as simple. The system will produce an error message in the event of a mistake.
- The visibility principle refers to all system features that are accessible through the UI. Users are not overrun by the number of options.
- The feedback principle: The design informs users of activities, mistakes, or exceptions through the use of messages.
- The reuse principle is to avoid ambiguity, the same names were used in the design process to carry out the same operations on various objects.
- Arrangement principle to put all important / urgent at screen corners using pop-up modal functions from WSS.

6.1.1 Web pages in a tree

1. From the “Login” page the registered public can reach the “dashboard” page. While, registered public can reach the “Register” page to create a new account.
2. From the login page the maintenance team can reach the “Dashboard” page. “Dashboard” page can reach “Manage sensor” and “Manage aquatank marine life”.
3. From the login page the management team can reach the “Dashboard” page. “Dashboard” page can reach the “Manage sensor”, “Manage aquatank water condition”, “Manage alarm” and “Analyse cost” page.
4. From the login page the Supplier can reach the “Dashboard” page. “Dashboard” page can reach “Supplier resources”.

These pages cover every aspect of the system's required functionality. The pages on this website are simple to navigate.

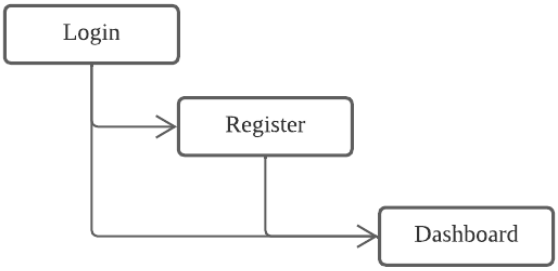


Figure 6.1.1.1 A tree of web pages for public

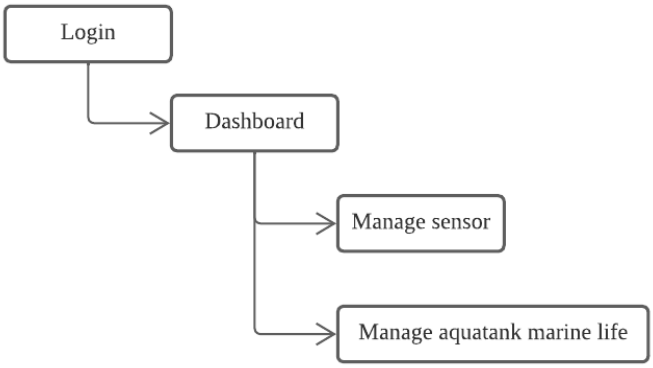


Figure 6.1.1.2 A tree of web pages for maintenance team

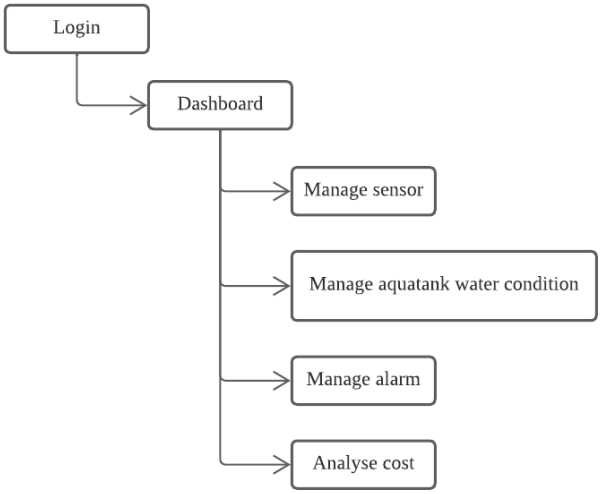


Figure 6.1.1.3 A tree of web pages for management team

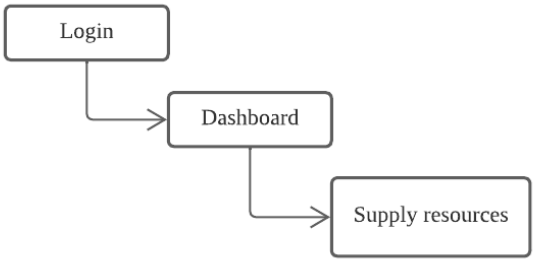


Figure 6.1.1.4 A tree of web pages for purchaser

6.1.2. User interface

Used GUI components are menus, buttons, text boxes, down drop lists, and tables. The only means of access to the entire database, by all actors, is through this UI.

6.1.2.1. Register UI

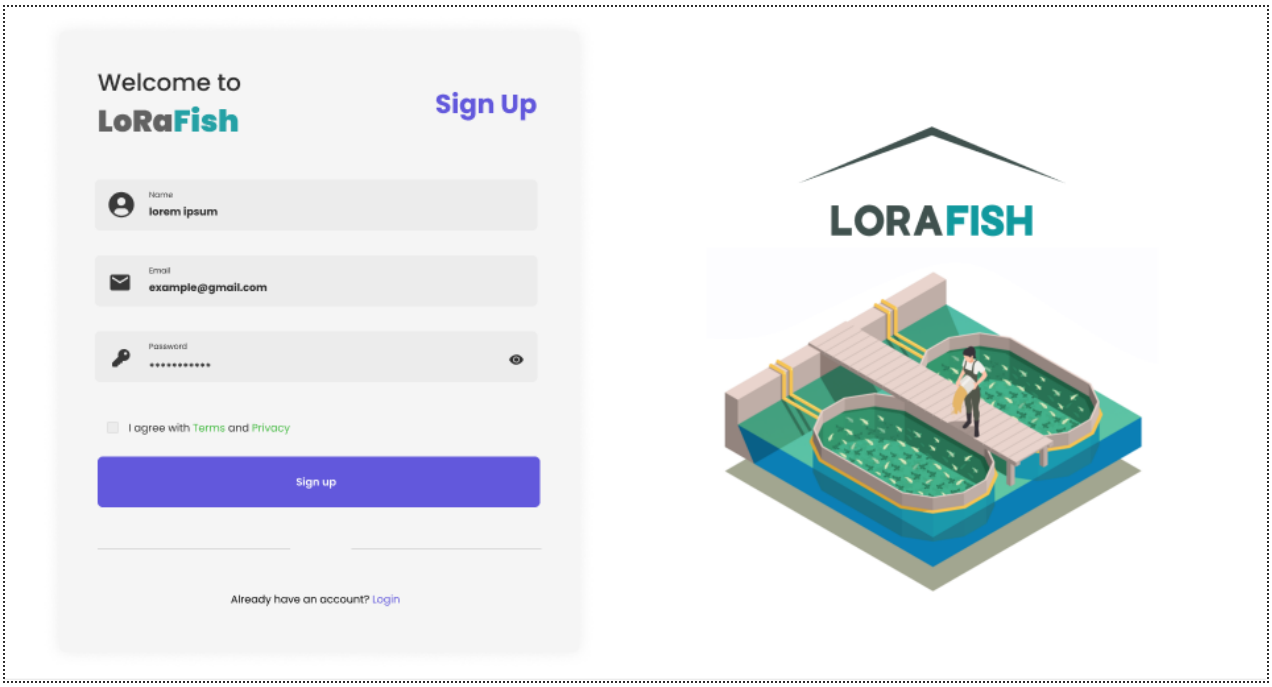


Figure 6.1.2.1 User Interface for Register

The Register UI is the user interface component responsible for allowing new users to create an account and access the LoraFish system. The Register UI requires the user to input their name, email address, and a password. The Register UI will validate user input by checking that the email address is in a valid format, and that the user has entered all required information. If any validation errors occur, the Register UI will display appropriate error messages to the user. Once the user has successfully registered, they will be redirected to the Login UI to access the system.

6.1.2.2. Login UI

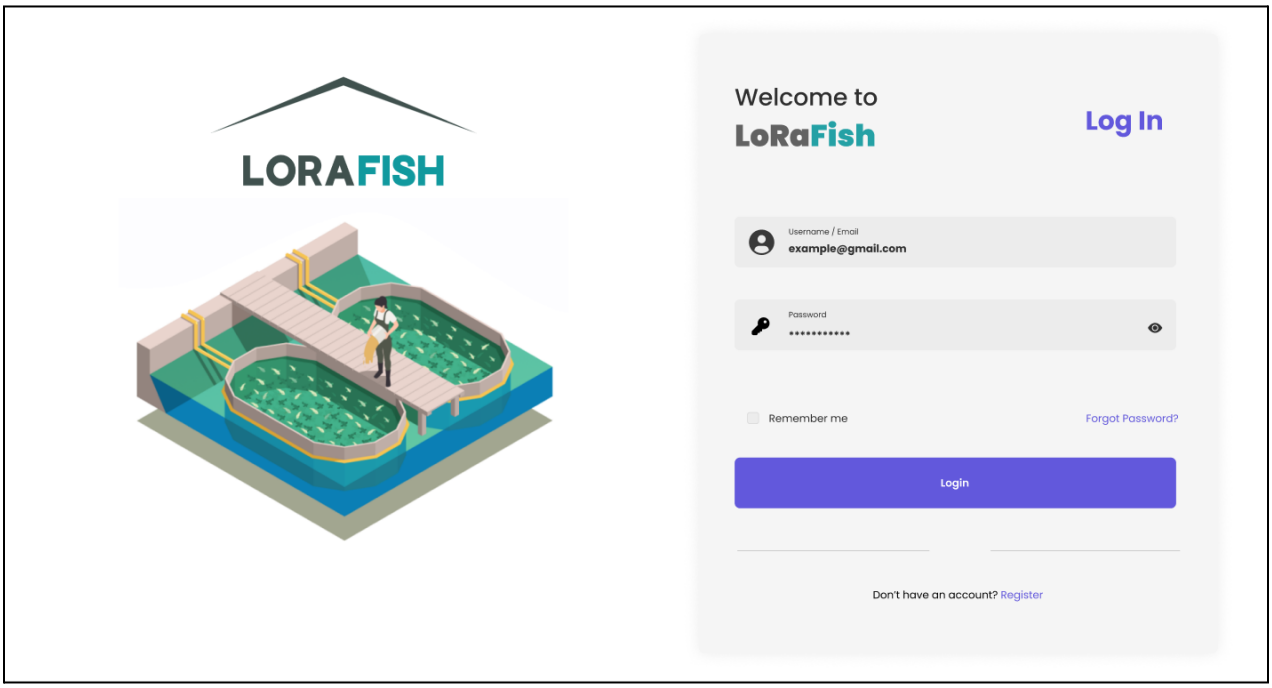


Figure 6.1.2.2. User Interface for Login

The Login UI is the user interface component responsible for authenticating users and allowing them to access the LoraFish system. The Login UI requires the user to input their username or email address and password. The Login UI will validate user authentication by checking the password matches the user's account information. If any validation errors occur, the Login UI will display appropriate error messages to the user. Once the user has successfully authenticated, they will be redirected to the Dashboard UI to view the relevant information and interact with the system. The Login UI should also include functionality for users to reset their password in case they forget it.

6.1.2.3. Dashboard UI

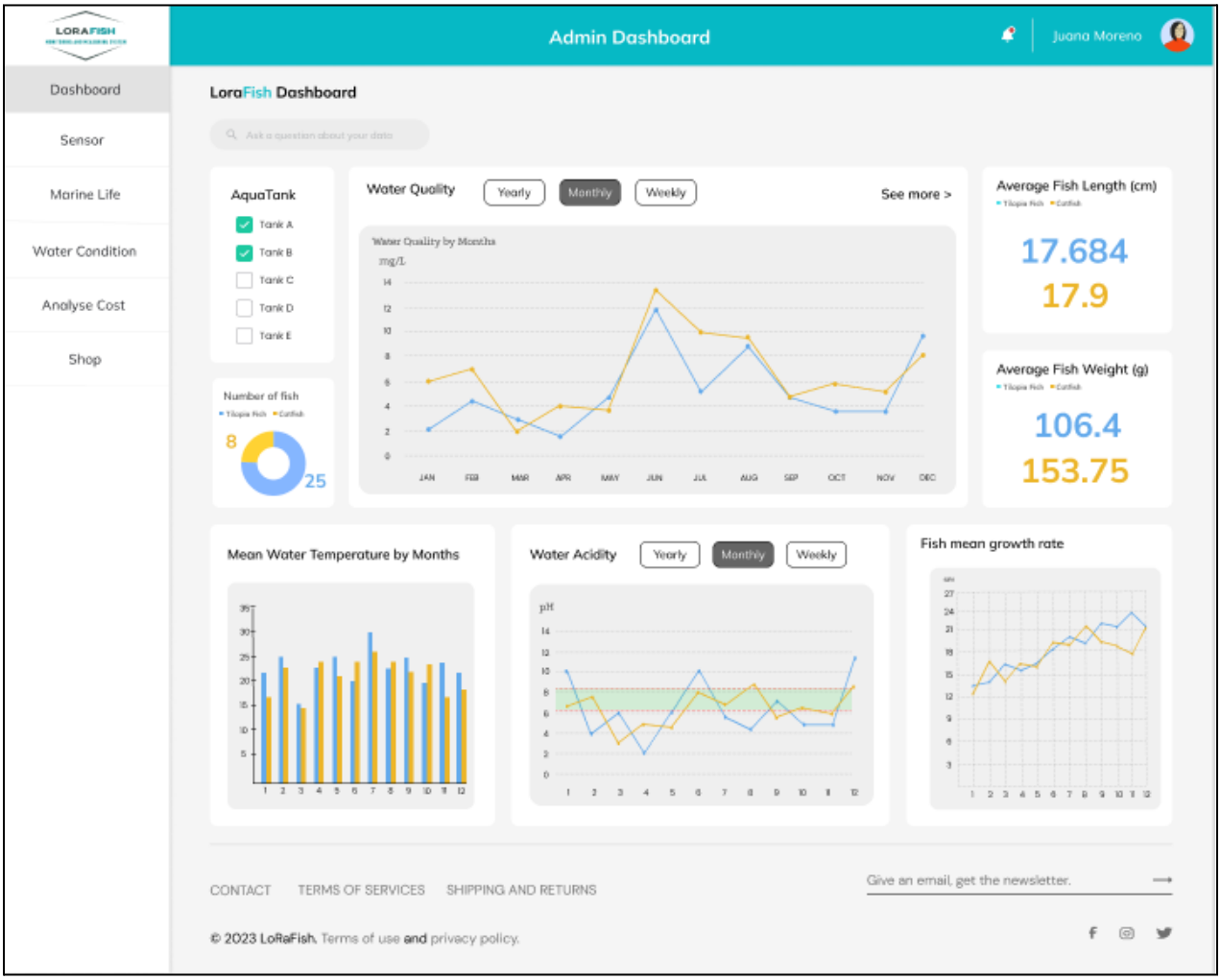


Figure 6.1.2.3.1 User Interface for Main Dashboard

The Dashboard UI is the user interface component responsible for displaying real-time information related to the aquatank, including water quality, average fish length and fish weight, water temperature and water acidity. The Dashboard UI will be tailored to different types of users, providing functionality based on their user roles (on the left hand side). For example, the Maintenance Team can manage sensors and marine life, the Management Team can manage water conditions and alarms and analyse costs, the Purchaser can purchase items for the aquatank if quality levels decrease, and the Public User can only view the dashboard. The Dashboard UI should also include functionality for users to customise their view, such as selecting which data to display and adjusting the time range for the displayed data.

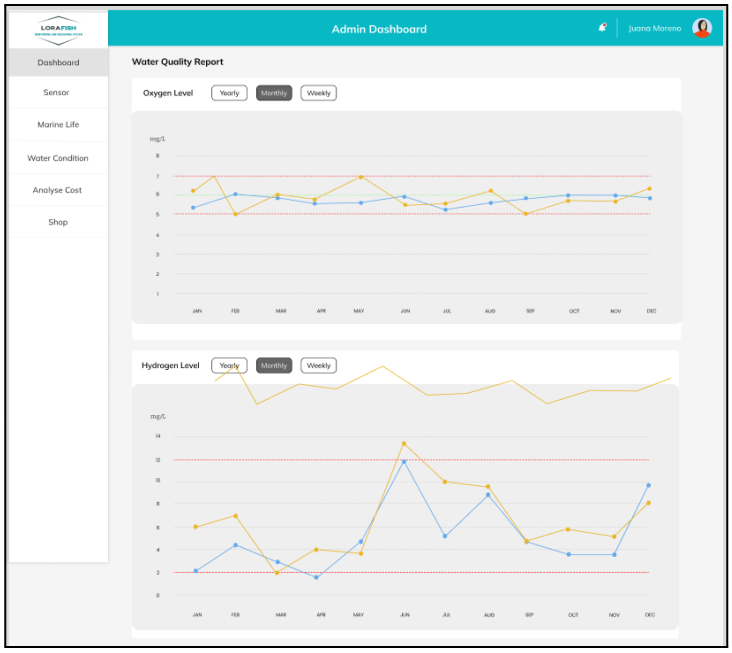


Figure 6.1.2.1.3.2(1) User Interface for Sensitivity analysis

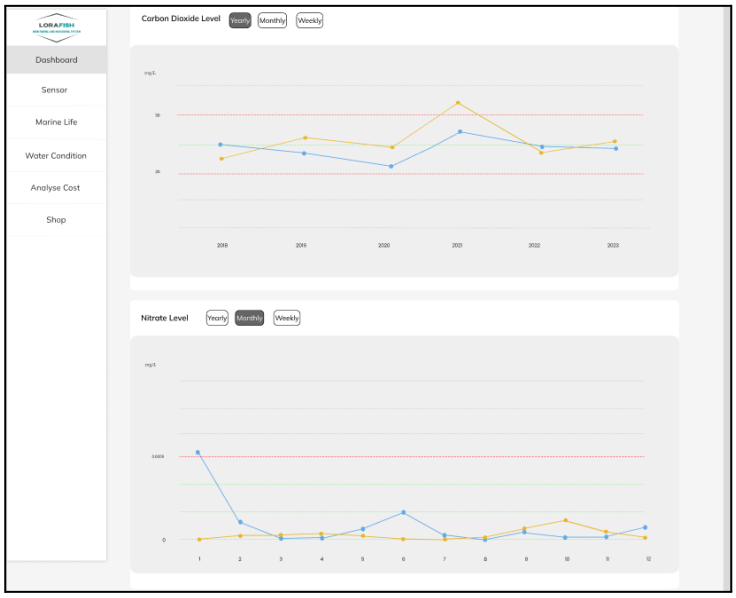


Figure 6.1.2.1.3.(2) User Interface for Sensitivity analysis

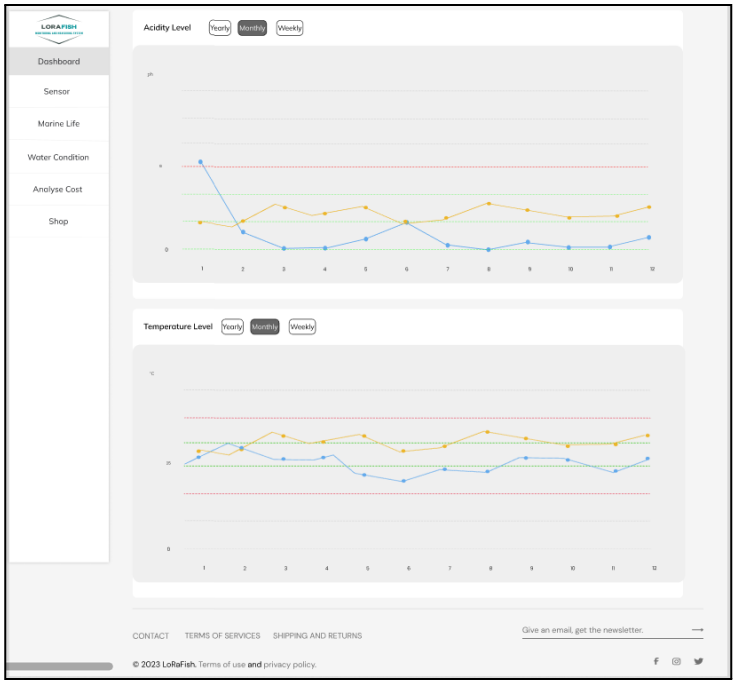


Figure 6.1.2.1.3.2(3) User Interface for Sensitivity analysis

The Water Quality Report is a user interface component that allows users to view detailed information about the water quality in the aquatank. This report can be accessed by clicking on the "Water Quality" section of the Dashboard UI. The Water Quality Report will display information such as oxygen level, hydrogen levels, carbon dioxide level, nitrate level, acidity level and temperature level. The user interface being referred to is equipped with advanced functionalities that allow the sorting of graphs using machine learning techniques. With this feature, users can easily arrange graphs according to their preferences, without the need for manual sorting. One of the most interesting features of this UI is its ability to sort graphs based on their values. In other words, the graph that has the highest number that exceeds the maximum or minimum range will be arranged in the first position, followed by the second-highest and so on. This ensures that the most critical data points are easily visible and readily accessible, enabling users to make informed decisions quickly and efficiently.

6.1.2.4. Manage Sensor UI

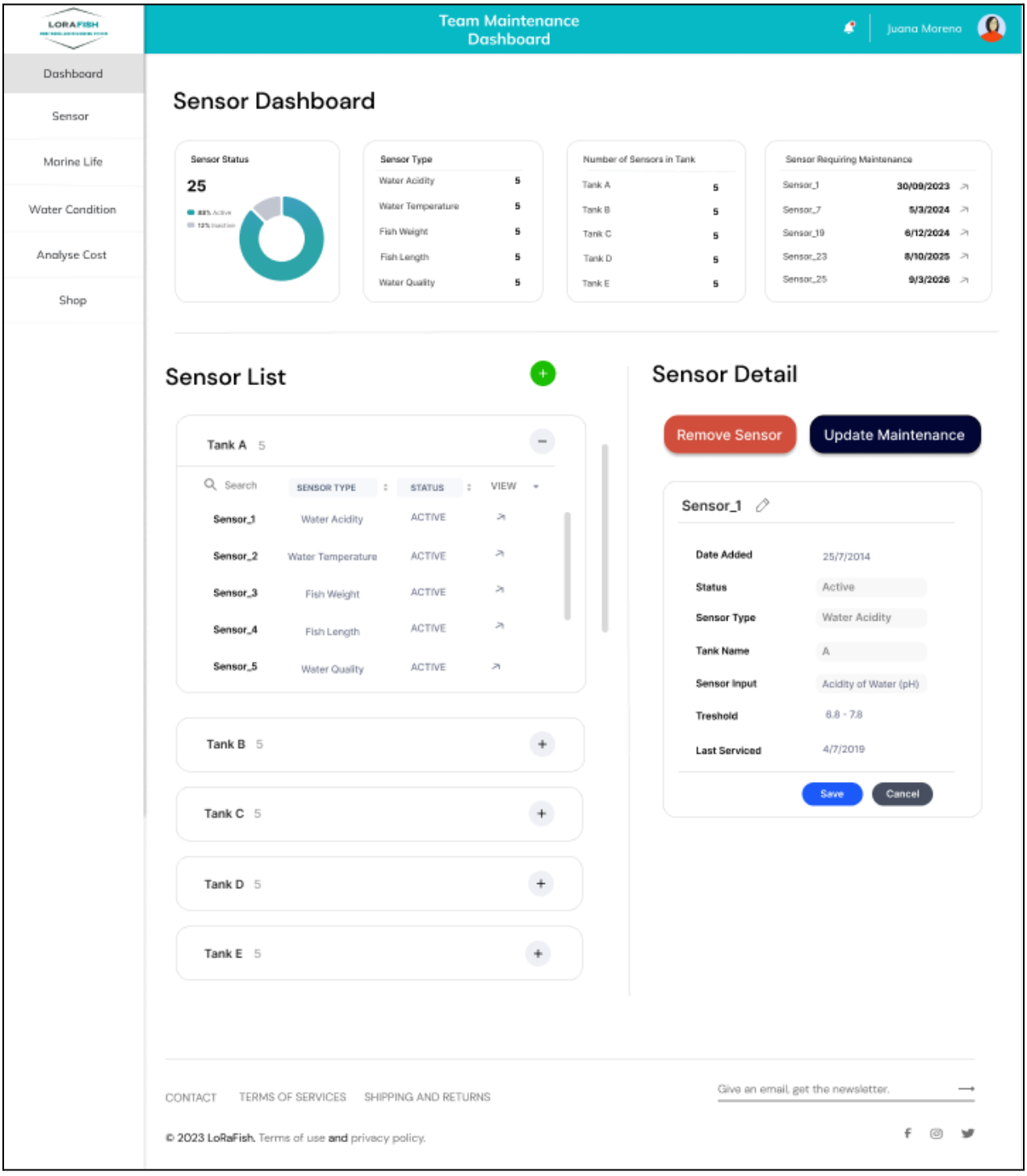


Figure 6.1.2.4.1 User Interface for Manage Sensor

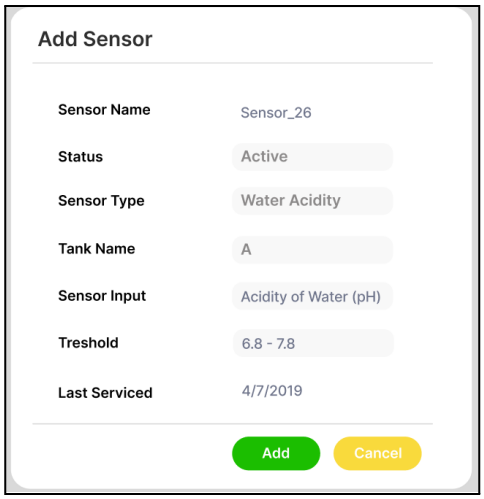


Figure 6.1.2.4.2 User Interface for Add Sensor

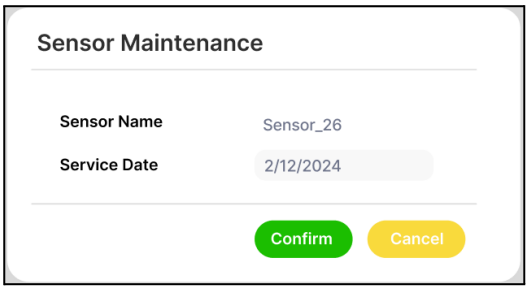


Figure 6.1.2.4.3 User Interface for Update Maintenance Date

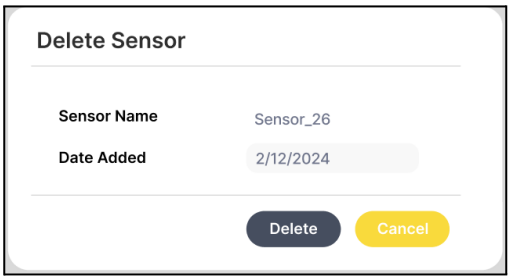


Figure 6.1.2.4.4 User Interface for Delete Sensor

The Manage Sensor user interface provides an intuitive and user-friendly way for the maintenance team to manage the sensors in the aquatank system. From this interface, the user can easily add new sensors to the system, remove sensors that are no longer needed, and view the status of each sensor in real-time. The Manage Sensor UI also provides useful information about each sensor, such as its unique identifier, location within the aquatank system, and the date of its last maintenance check.

6.1.2.5 Manage Aquatank Marine Life UI

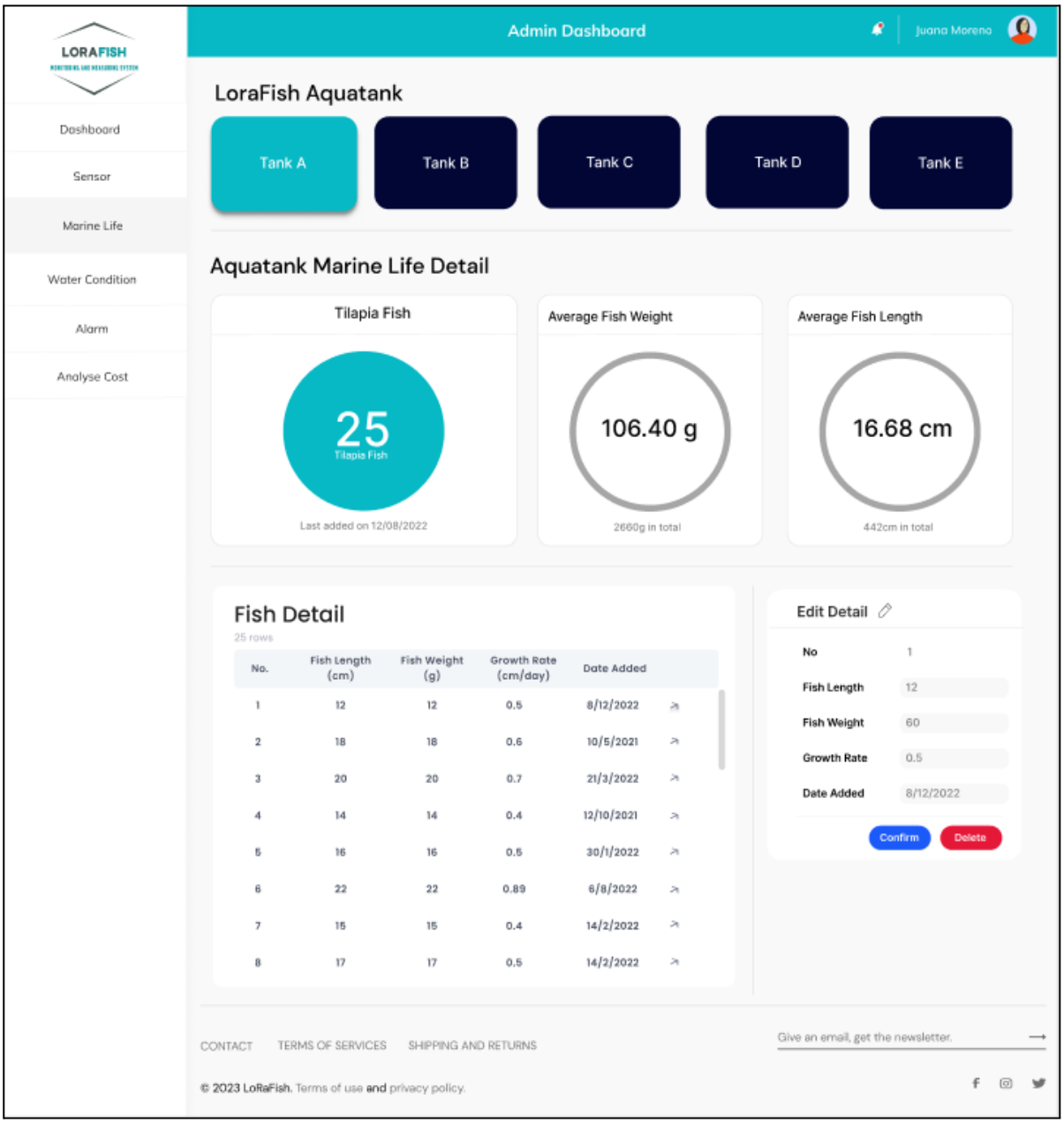
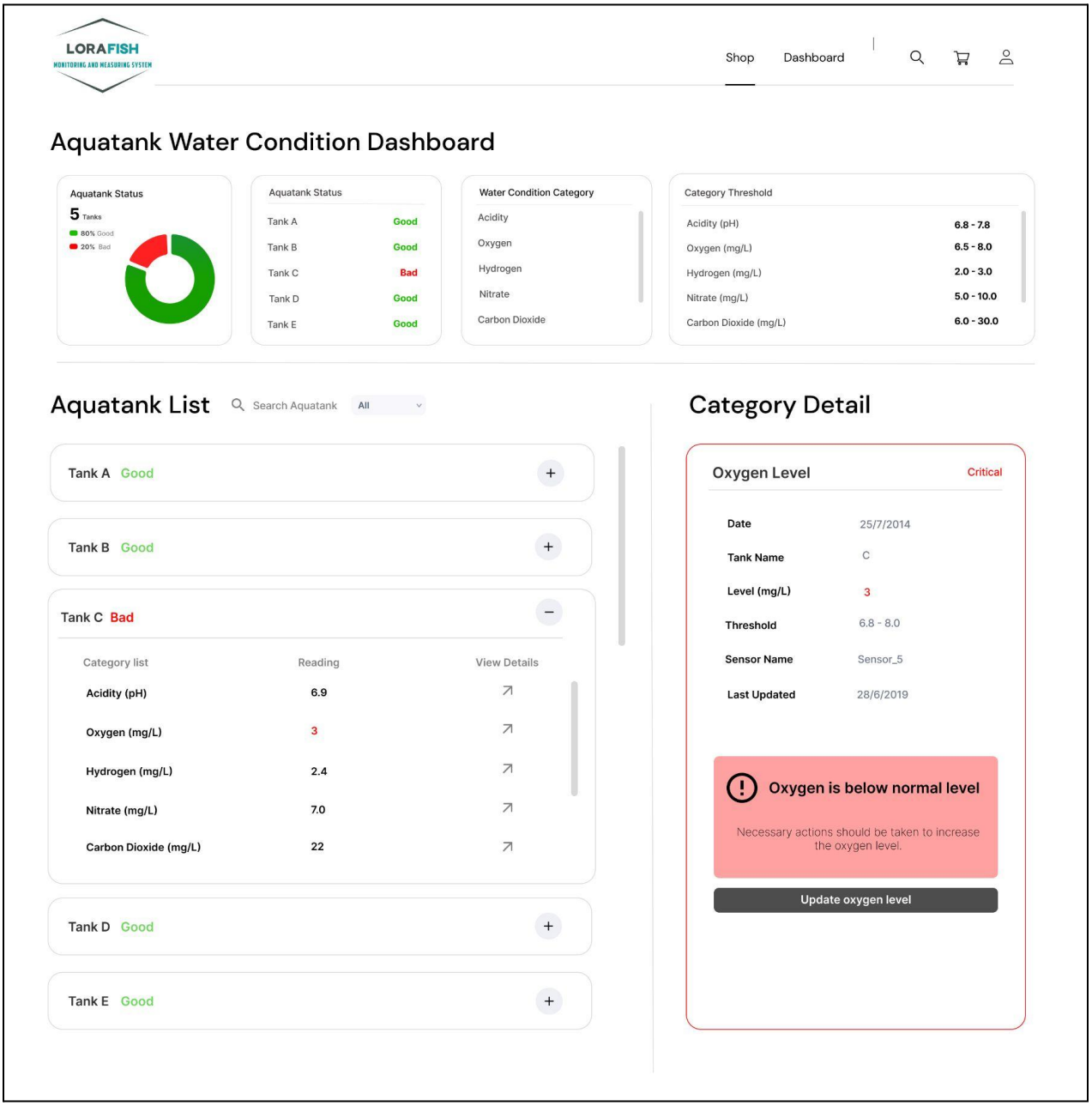


Figure 6.1.2.5 User Interface for Manage Aquatank Marine Life

The Manage Aquatank Marine Life user interface provides an interactive platform for users to view, edit, and manage data related to marine life in the aquatank. Additionally, the interface helps users better understand trends and patterns in the aquatank's marine life population.

6.1.2.6. Manage Aquatank Water Condition UI



Manage aquatank water condition user interface is designed to display the acidity levels, gases content, and temperature of the aquatank. The interface also listed the set threshold for each category. An alert is displayed if the reading exceeds the set threshold.

6.1.2.7. Manage Alarm UI

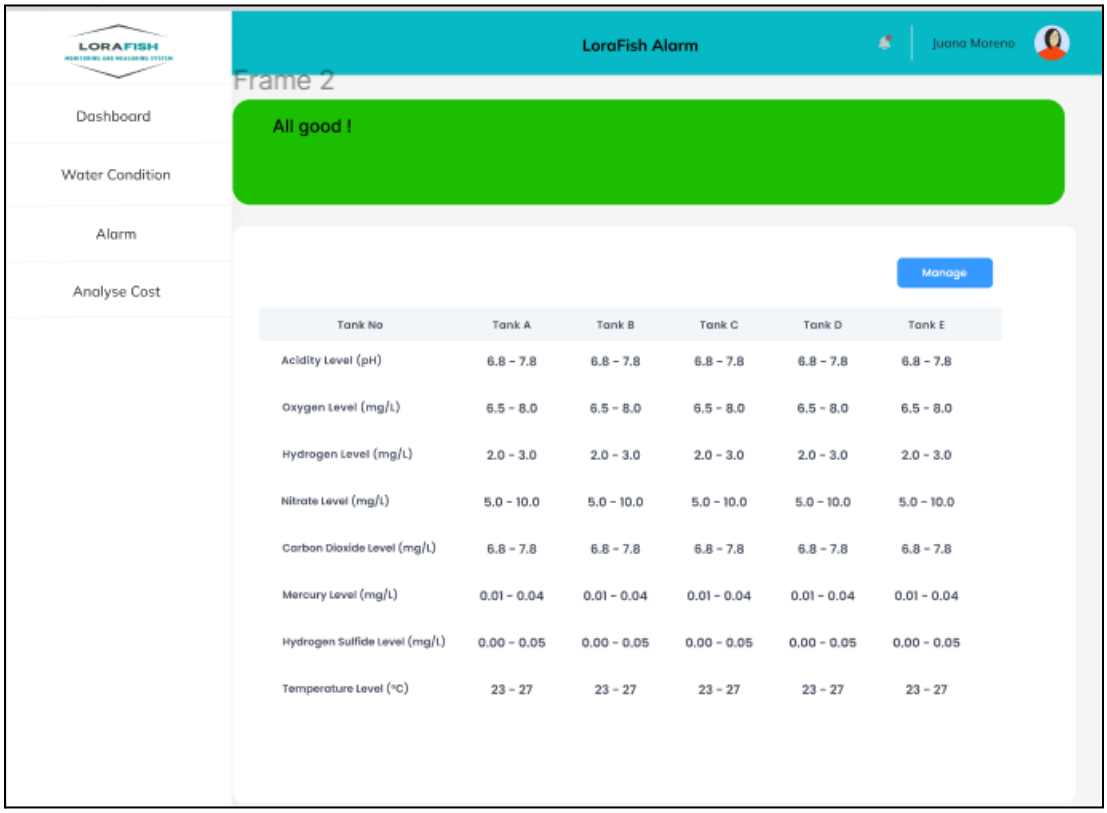


Figure 6.1.2.7.1 : Display good condition water aquatank

The user interface page designed for monitoring the condition of an aquatank's water is user-friendly and visually appealing. The page displays key metrics such as temperature, pH level, dissolved oxygen level, hydrogen level, nitrate level, carbon dioxide level, mercury level and hydrogen sulphide level in an easy-to-read format. One of the standout features of this user interface page is the "manage" button. When clicked, this button takes the user to a separate page where they can edit the maximum and minimum values for each metric.

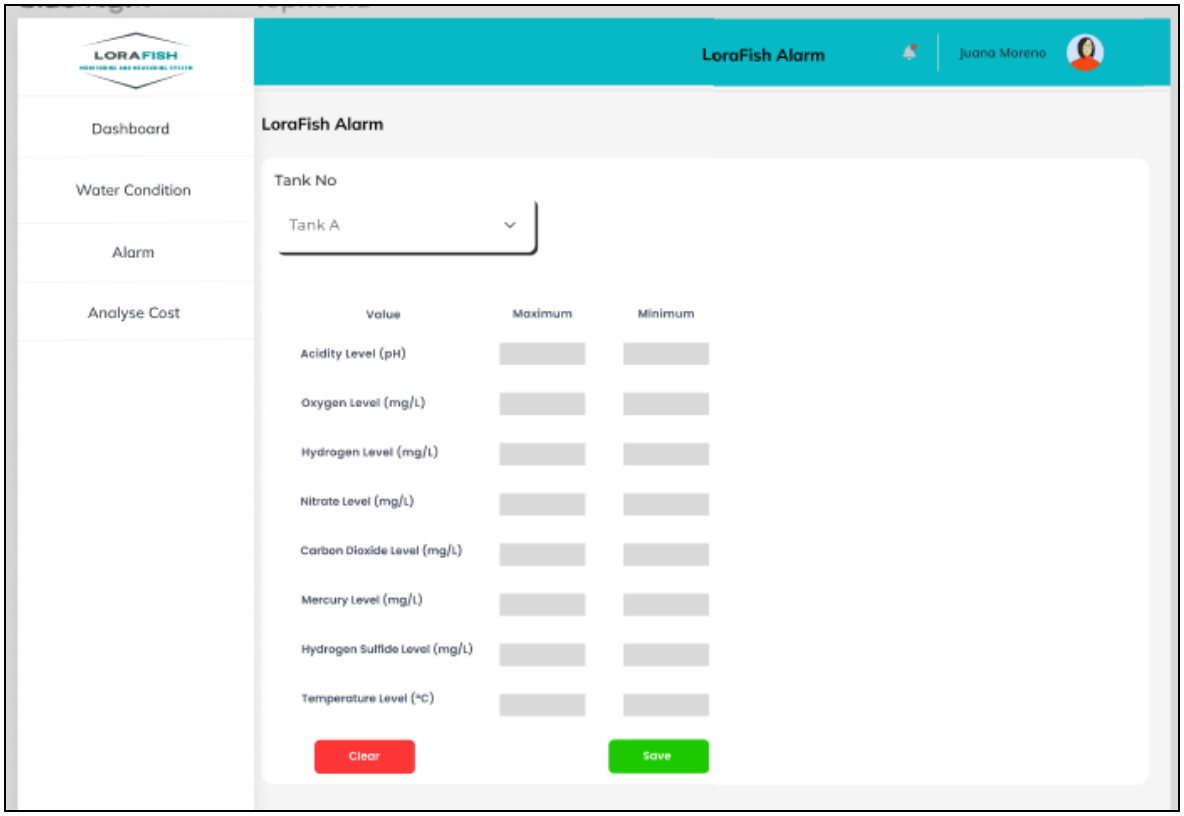


Figure 6.1.2.1.7.2 : Manage good condition of aquatank water

This customization feature ensures that the user can set the specific parameters that they deem optimal for their aquatank's water condition, giving them complete control over the monitoring process. Overall, this user interface page is a valuable tool for those seeking to maintain a healthy and stable environment for their fish life.

6.1.2.8. Analyse Cost UI

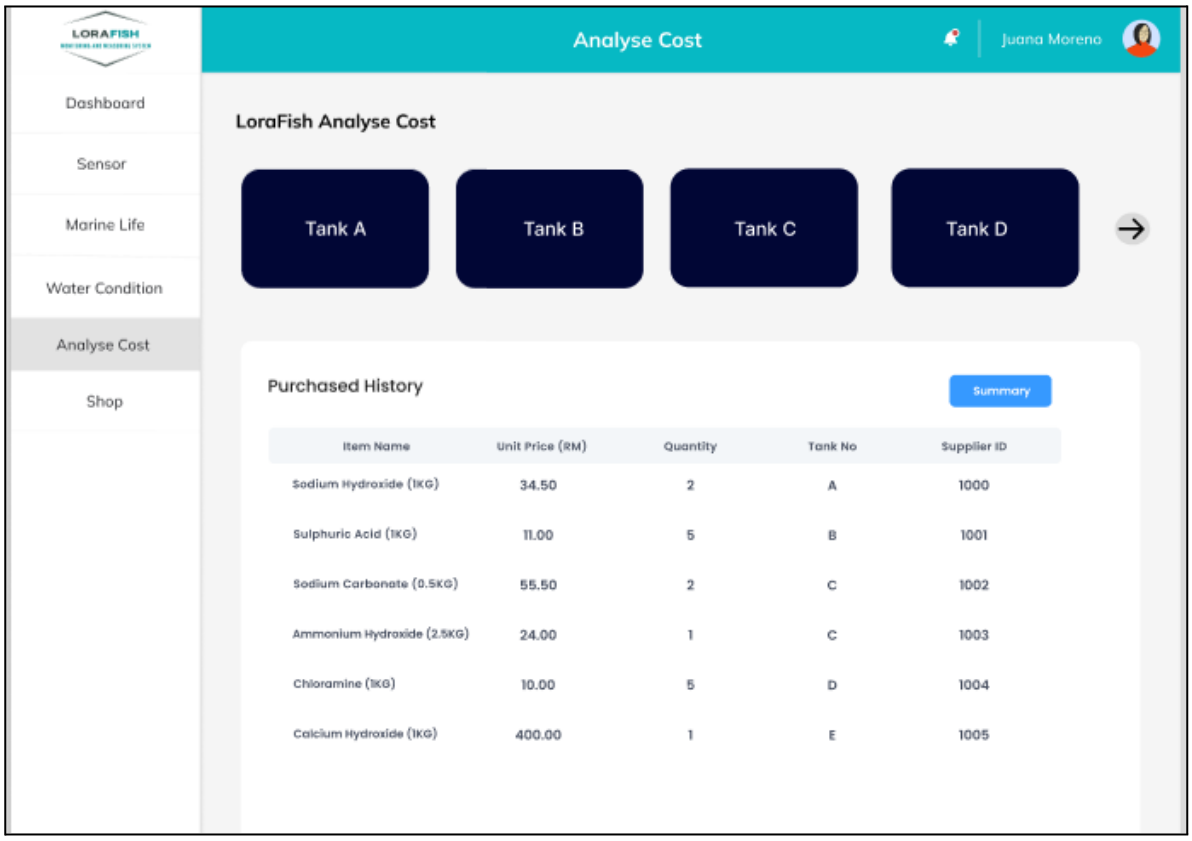
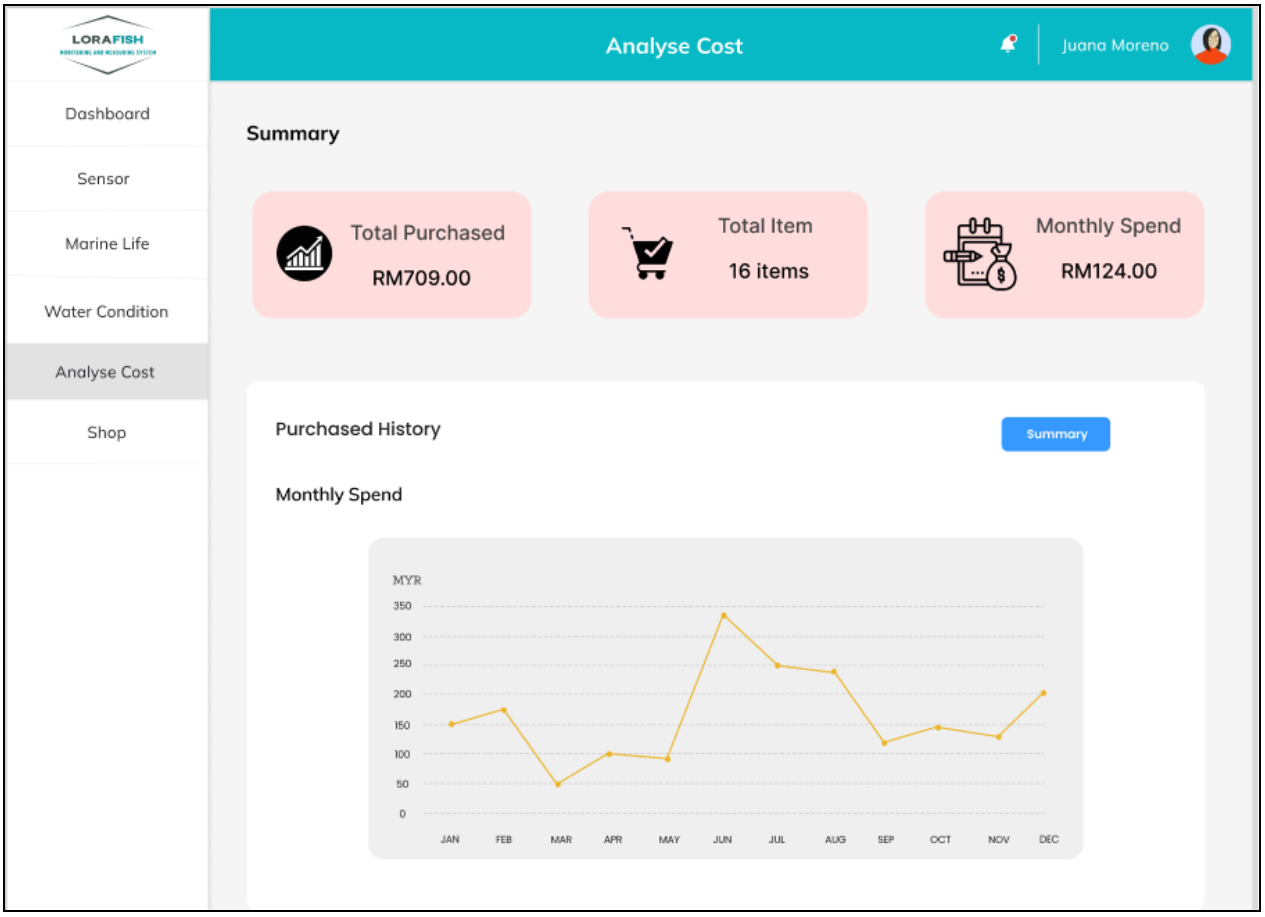


Figure 6.1.2.1.8.1 : Analyse Cost UI

Analyse Cost Interface provides users with detailed information about the cost of various items. The interface displays the attributes of each item, such as item name, unit price, quantity, tank number, and supplier ID.



A summary page that displays a graph for monthly spend is a useful tool for tracking expenses over time. The graph typically shows the amount of money spent in a month, with each month represented as or line on the graph. The purpose of this display is to provide a visual representation of spending trends, making it easy to identify any fluctuations or patterns in spending.

6.1.2.9. Supply Resources UI

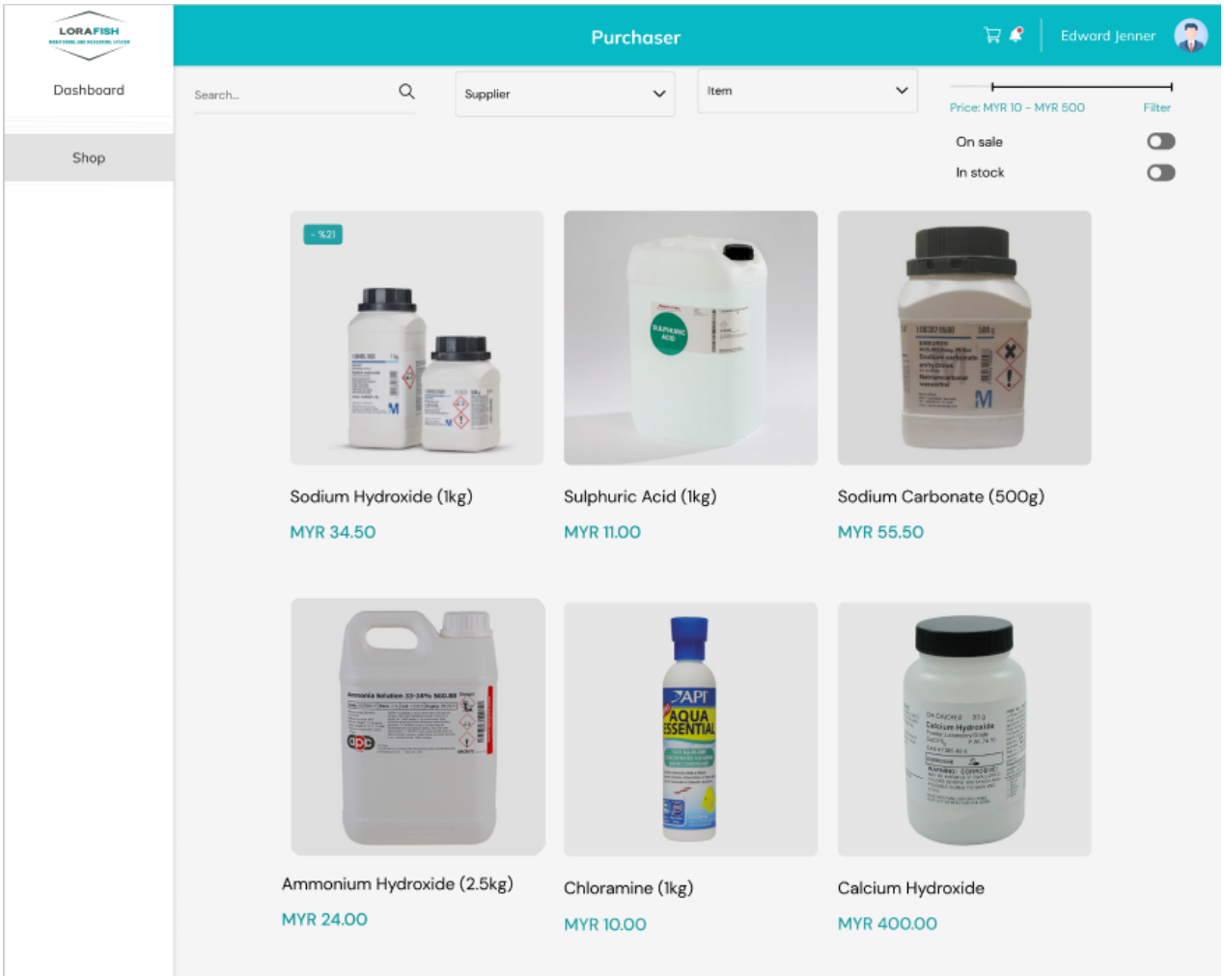


Figure 6.1.2.1.9: Supply Resources User Interface

Supply Resources User Interface provide user (purchaser) to search any chemical substances that they needed to neutralise the acidity level of aquatank and maintain the level of water quality in the aquatank. The user also can track their purchased history to track the previous supplier.

7. Class diagrams

7.1. Full Class Diagram

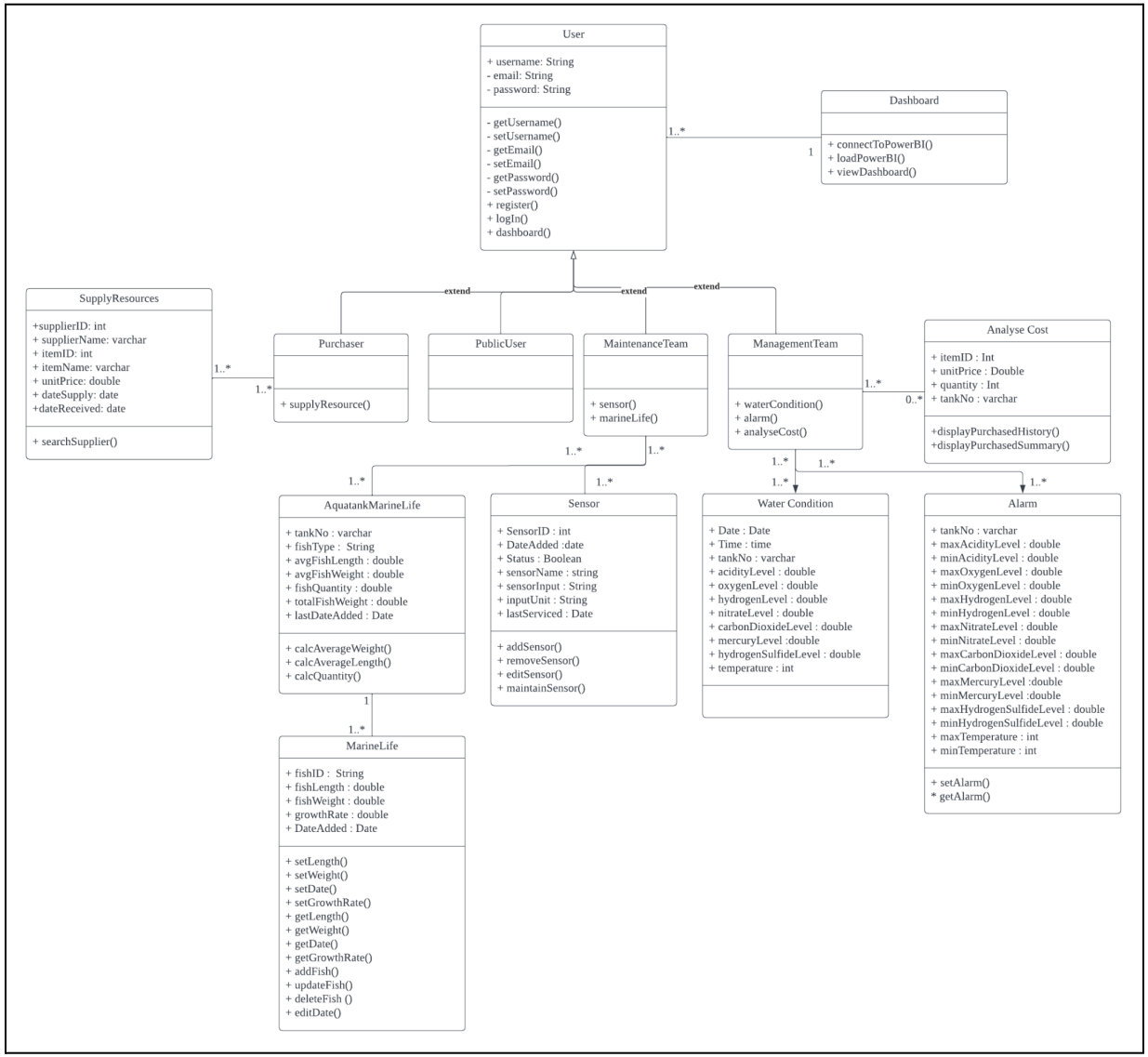


Figure 7.1 Full class diagram

BIBLIOGRAPHY

Abstract Factory Pattern, UML diagram, Java Example. In *A&P Web Consulting.* Retrieved Mar. 25, 2010, from http://www.apwebco.com/gofpatterns/creational/AbstractFactory.html

Advantages and Disadvantages of Collection Framework. In *RoseIndia*. Retrieved Mar. 25, 2010, from http://www.roseindia.net/java/jdk6/collection-advantagesdisadvantages.shtml

Design Pattern Variations: A Better Visitor. In *InformIt*. Retrieved Mar. 26, 2010, from http://www.informit.com/articles/article.aspx?p=1278987

Googles SkipFish – Web App Security Scanner. In *a4apphack*. Retrieved on March 28th 2010 from http://a4apphack.com/index.php/featured/googles-skipfish-webapp-security-scanner

IEEE 1016-1.pdf. In *essex*. Retrieved Mar. 25, 2010, from http://courses.essex.ac.uk/CE/CE201/restricted/projectDocuments/ieee1016-1.pdf

J2SE. In *Sun developer network*. Retrieved Mar. 26, 2010, from http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Exception.html

MCV for beginners. In chtivo. Retrieved Apr. 5, 2010, from http://chtivo.webhost.ru/articles/mvc.php

Samples. In *Concordia University*. Retrieved Feb. 3 and 13, 2010, from http://users.encs.concordia.ca/~c55414/samples/

SDD. In *cmcrossroads*. Retrieved Mar. 25, 2010, from

http://www.cmcrossroads.com/bradapp/docs/sdd.html SDD. In *pdfqueen.*

Retrieved Mar. 25, 2010, from

http://www.pdfqueen.com/html/aHR0cDovL3dpbG1hLnZ1Yi5hYy5iZS9+c2UxXzA4M

DkvZG9jdW1lbnRzL2ZpcnN0Y3ljbGVfY29weS9TREQucGRm

Skipfish. In *Google*. Retrieved on March 25, 2010, from http://code.google.com/p/skipfish/

Skipfish: An Active Web Application Security Reconnaissance Tool! In *PenTestIT*. Retrieved on March 27, 2010 from http://pentestit.com/2010/03/20/skipfishactive-web-application-security-reconnaissance-tool/

SkipfishDoc. In *Google*. Retrieved on March 25, 2010 from http://code.google.com/p/skipfish/wiki/SkipfishDoc

Skipfish Updates In *Google*. Retrieved on April 12th, 2010 from http://code.google.com/p/skipfish/updates/list

Std_public. In *IEEE Standards Association*. Retrieved Mar. 25, 2010, from http://standards.ieee.org/reading/ieee/std_public/description/se/1016-1998_desc.html

User Interface Design Tips, Techniques, and Principles. In *ambysoft*. Retrieved Apr. 5, 2010, from http://www.ambysoft.com/essays/userInterfaceDesign.html

Visitor Pattern Versus Multimethods. In *Sourceforge.* Retrieved Mar. 26, 2010, from http://nice.sourceforge.net/visitor.html