**Wed, Sept 16, 2016**

**PiSeas**

**BTS530 - Major Project, Planning & Design**

**Professor Barb Czegel**

**Group 7**

**Matt Babol**

**Van Bui**

**Sallie Jiang**

**Michael Michalski**

Table of Contents

[BTR490 System 2016 3](#_Toc467139053)

[System Name and Team Number 3](#_Toc467139054)

[Team Members 3](#_Toc467139055)

[Client and/or Mentor 3](#_Toc467139056)

[System Overview 3](#_Toc467139057)

[Technology to be used 3](#_Toc467139058)

[Raspberry Pi (Raspberry Pi 3 Model B) 3](#_Toc467139059)

[Open Aquarium Shield (Runs on Raspbian OS) 4](#_Toc467139060)

[Open Aquarium Sensors 4](#_Toc467139061)

[Mobile Angular UI 4](#_Toc467139062)

[Xamarin 4](#_Toc467139063)

[Constraints: 4](#_Toc467139064)

[Stakeholders 5](#_Toc467139065)

[A Diagrammatic Representation of the System 5](#_Toc467139066)

[Feeding Sub-System 6](#_Toc467139067)

[Tank 20](#_Toc467139068)

[Water 28](#_Toc467139069)

[Lighting Sub-System 53](#_Toc467139070)

[Temperature Sub-System 68](#_Toc467139071)

# BTR490 System 2016

## System Name and Team Number

Automated Fish Tank with Raspberry Pi (Pi-Seas) - Team # 7

## Team Members

Matthew Babol

Van Chau Bui

Sallie Jiang

Michael Michalski

## Client and/or Mentor

Mentor: Diego Flores, Store Manager at Big Al’s Pet Supercenters

## System Overview

The system we propose is a prototype for an automated fish tank, created with the incorporation of the Raspberry Pi. The Raspberry Pi will be programmed to control all the use cases and tasks required to sustain a habitable fish tank environment. Some functionalities of the tank include scheduled feedings, water temperature control, water changing, camera monitoring, light control, and water analysis. The data obtained from the built-in sensors within the tank is made available on the user’s phone, and accessed through an app which we will also design and create. The app is able to connect to multiple fish tanks and will be able to monitor and control each individual tank in real time through Bluetooth. The user is able to monitor tank information as well as schedule tasks for the system in order to maintain an optimal aquatic habitat.

## Technology to be used

### Raspberry Pi (Raspberry Pi 3 Model B)

Raspberry Pi is a series of single-board computers ranging from $20-$60 CND. The version of Raspberry Pi we are looking at includes 1GB of RAM, ARM compatible CPU, GPU, MicroSDHC memory storage, 4 USB ports, Ethernet port, as well as WiFi and Bluetooth capabilities. The Raspberry Pi uses a Linux-Kernel based OS, Raspbian.

The Raspberry Pi will act as our central processing unit, controlling all other components of the tank itself. The Raspberry Pi does not have a real-time clock built in, and will need to be reset after each power off. Raspberry Pi runs on Python natively, however we would like to use C++ to take advantage of the Open Aquarium API. While the Raspberry Pi does not run on Windows Systems, we plan to create an app for Android devices to communicate with our system.

### Open Aquarium Shield (Runs on Raspbian OS)

Shields are extension boards at are easily mountable to Raspberry Pi units. The Open Aquarium Shield is a sensor platform designed to work with aquatic system sensors. It is based on Arduino and is backed by an open source API which we will use to bridge all other aspects of our system with the Raspberry Pi. The API provided is in C++, if we choose to code in Python we will have to recreate some features of their API.

### Open Aquarium Sensors

There are various sensors that will provide information to the Pi-Seas system. The system will need a water temperature sensor, vertical liquid level sensor, conductivity sensor, pH sensor and a filter quality sensor. These components are made compatible with the Raspberry Pi through the Open Aquarium Shield. The use of these sensors are explained in section 8 through use cases specifications.

### Mobile Angular UI

Mobile Angular UI is a HTML 5 framework that uses Bootstrap 3 and Angular JS. We hope to use this to improve the usability of our app. This framework uses libraries such as fastclick.js to provide smooth and responsive user experience. With our background knowledge of Bootstrap and JavaScript, it is easy to learn and implement.

### Xamarin

Xamarin is a mobile development framework designed to create native apps for Windows, iOS and Android in C#. Xamarin would provide a consistent user experience for all users across many platforms. We will be mainly focusing on developing an Android app as the interface to the Pi-Seas system.

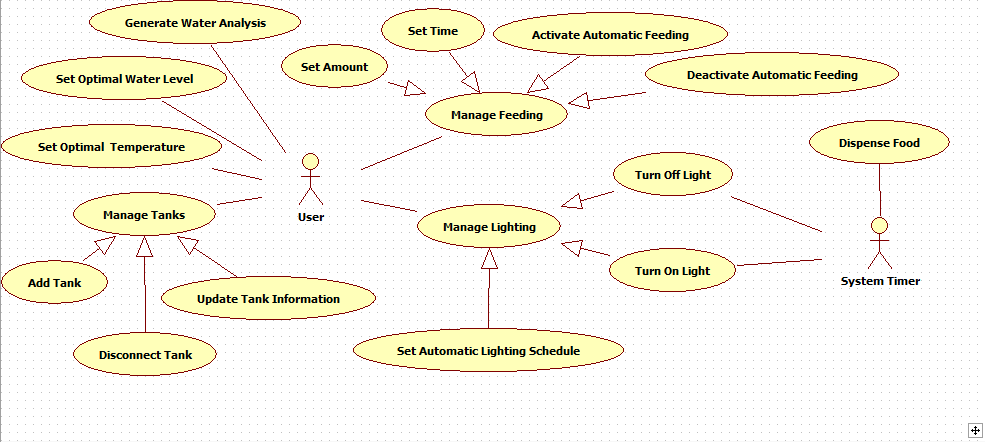
### Constraints:

* Mobile app must run on Android.
* Mobile device must be able to connect Raspberry Pi 3 Model B via Bluetooth v4.0
* The Open Aquarium shield can only support 8 different sensors/hardware components
* Tank size must between 20 to 33 gallons due to heater limitations (100W)
* Pi-Seas must be powered by an external power supply of 12V - 2A

## Stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Team 7 | Technical Developer |
| Barb Czegel | Authority |
| Diego Flores | Business Domain Expert |
| Big Al’s | Advisor |
| Tank/Fish Owners | End Users, Customers |
| Pet Store Employees | End Users |
| PETA | Onlooker |

## A Diagrammatic Representation of the System



### Feeding Sub-System

The system will automatically feed the aquatic animals based on the settings chosen by the user. The settings include the amount of feed and timing. The settings will be accessible through the mobile app interface. Manual feeding can also be chosen.

#### Use Case Name: Enable automatic feed

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page
* Feeder must be connected to the tank system
* Automatic feed must be disabled

**Use Case Successful post condition(s)**:

* Automatic feeding is enabled

**Applicable Business Rules:**

* Fish can only be fed twice a day
* Gap between feeds must be greater than 2 hours

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Select enable automatic feed. | System checks if there is an existing feeding schedule. If a valid schedule exists, automatic feeding is enabled. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | No schedule existing. | Notify the user that there is no existing schedule made, ask user for create one first. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic feed is turned on. | Automatic feed is turned on, and will start feeding the fish during the scheduled times. |
| Alt: No existing schedule, create new schedule. | No existing schedule is made; user is directed to the “create new” schedule section. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic feed is enabled. | Valid schedule is already created. | Fish will be fed during scheduled times. |
|  | Alt: No existing schedule is made, choose to create new. | No existing feeding schedule is made. | User is directed to the “create new” schedule page. |

#### Use Case Name: Disable automatic feed

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page
* Feeder must be connected to the tank system
* Automatic feed must be enabled

**Use Case Successful post condition(s)**:

* Automatic feeding is disabled

**Applicable Business Rules:**

* Fish can only be fed twice a day
* Gap between must be greater than 2 hours

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Select disable automatic feed. | Confirm with if the user wants to disable the feeding. |
|  | Confirm disabling the feeding. | Automatic feeding is disabled. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User cancels disabling the automatic feeding. | User cancels, automatic feeding is left on. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic feed is turned off. | Automatic feed is turned off, and fish will not be fed automatically. |
| Alt: User cancels during confirmation. | Automatic feed is not turned on, go back to the feeding page. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic feed is disabled. | User confirmation is checked. | User confirms the disabling, and automatic feed is turned off. |
|  | Alt: User does not confirm. | User does not confirm disabling the automatic feed. | System will redirect the user to the tank feed page. System will not save any changes. |

#### Use Case Name: Add feeding schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page

**Use Case Successful post condition(s)**:

* A valid feeding schedule is created

**Applicable Business Rules:**

* Can have multiple schedules
* Cannot create a schedule with more than 2 feeds per day
* Feedings must be at least 2 hours apart

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | User selects the “add new” schedule option. | System opens up the “add new” schedule page. |
| 2 | User creates new schedule and saves. | System checks whether the new schedule will cause any days to have more than 2 feeds per day, and whether there are at least 2 hour gaps between feedings. If both the requirements are good, a new feeding schedule if added and shown in the feeding page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Feeding more than twice a day. | One or more days is scheduled for feeding more than twice a day, schedule is not saved, error message. |
| A2 | Gap between feeding does not exceed 2 hours. | The gap between feeds in one or more days does not exceed 2 hours, schedule is not saved, error message. |
| A3 | User cancels. | User cancels the activity and goes back, schedule is not saved. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successfully created schedule. | Schedule is successfully created and displayed in the feeding page. |
| Alt: Feeding exceeds twice a day. | Fish cannot be fed more than twice during each day. |
| Alt: Feeding gap under 2 hours. | Gap between feedings must be greater than 2 hours. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: New schedule created successfully. | Adequate feeding times per day, gap between feedings over 2 hours. | Feeding schedule created, new schedule is added into the main feeding page. |
|  | Alt: Feeding exceeds twice a day. | Feeding per day exceeds the maximum of 2 per day, gap between feedings over 2 hours. | Feeding schedule not made, no new feeding schedule shown in the main feed page. Error message: “Only 2 feeds per day allowed”; options displayed. |
|  | Alt: Gap between feedings does not exceed 2 hours. | Adequate feeding times per day, gap between feedings does not exceed 2 hours. | Feeding schedule not made, no new feeding schedule shown in the main feed page. Error message: “Must have a 2-hour gap between feedings”; options displayed. |

#### Use Case Name: Delete feeding schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page
* Must have at least one feeding schedule

**Use Case Successful post condition(s)**:

* Feeding schedule is deleted

**Applicable Business Rules:**

* Automatic feeding will be disabled if schedule is deleted

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | User selects which schedule to delete. | System confirms with the user if they want this schedule to be delete. |
| 2 | User confirms deletion. | System deletes. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User cancels. | One or more days is scheduled for feeding more than twice a day, schedule is not saved, error message. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Schedule is successfully deleted. | Schedule is successfully deleted and removed from the main feeding page. |
| Alt: User cancels during confirmation. | Automatic feed is not enabled, go back to the feeding page. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Schedule is successfully delete. | User accepts deletion confirmation. | Feeding schedule is deleted, schedule is removed from the main feeding page. Feeding Automation will be disabled. |
|  | Alt: System is unable to delete feeding schedule as the user does not accept changes. | User does not confirm deleting the schedule. | System will redirect the user to the tank feed page. System will not save any changes. |

#### Use Case Name: Update feeding schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page
* A valid feeding schedule must be created

**Use Case Successful post condition(s)**:

* Feeding schedule is updated

**Applicable Business Rules:**

* Cannot have a schedule with more than 2 feeds per day
* Feedings must be at least 2 hours apart
* Can have multiple feeding schedules

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | User selects which schedule to update. | System opens up the “edit schedule” for the selected feeding schedule. |
| 2 | User updates the current schedule. | System checks whether the new schedule will cause any days to have more than 2 feeds per day, and whether with the new schedule will have at least 2 hour gaps between feeds. If both the requirements are good, the system asks the user for the final confirmation. |
| 3 | User confirms the updated schedule. | The schedule is updated, and is shown on the main feeding page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Feeding more than twice a day. | One or more days is scheduled for feeding more than twice a day, schedule is not saved, error message. |
| A2 | Gap between feeding does not exceed 2 hours. | The gap between feeds in one or more days does not exceed 2 hours, schedule is not saved, error message. |
| A3 | User cancels. | User cancels the activity and goes back, schedule is not saved. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successfully created schedule. | Schedule is successfully updated and displayed in the feeding page. |
| Alt: Feeding exceeds twice a day. | Fish cannot be fed more than twice during each day. |
| Alt: Feeding gap under 2 hours. | Gap between feedings must be greater than 2 hours. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Schedule updated successfully. | Adequate feeding times per day, gap between feedings over 2 hours. | Feeding schedule is updated, new schedule times are updated on the main feeding page. |
|  | Alt: Feeding exceeds twice a day. | Feeding per day exceeds the maximum of 2 per day, gap between feedings over 2 hours. | Feeding schedule not updated. Days that have more than 2 feeds are marker on the page. Error message: “Only 2 feeds per day allowed”; options displayed. |
|  | Alt: Gap between feedings does not exceed 2 hours. | Adequate feeding times per day, gap between feedings does not exceed 2 hours. | Feeding schedule not updated. Days where feeding does not exceed 2 hours are marker on the page, with the other scheduled times shown. Error message: “Must have a 2-hour gap between feedings”; options displayed. |

#### Use Case Name: Manual feeding

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Feeding” mobile app page
* Feeder must be connected to the tank system

**Use Case Successful post condition(s)**:

* Fish are fed

**Applicable Business Rules:**

* Fish can only be fed twice a day regardless of manual or automatic
* Gap between must be greater than 2 hours
* User can adjust feeder for a specific amount of food to be fed

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Select manual feeding. | System checks if any food is available, checks if fish have been fed more than twice in a day, and checks if gap between feedings have been under 2 hours. If all checks passed, fish are fed. Time of feeding logged. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | No food available. | Notify user that there is no more food in the feeder, end use case. |
| A2 | Fish have been fed within the last 2 hours. | Notify user that they cannot feed fish within a 2-hour time span, end use case. |
| A3 | Fish have been fed twice in the day already. | Notify user that system cannot feed fish more than twice a day, end use case. |
| A4 | Two feeds are scheduled for the day. | Notify user that system cannot feed fish more than twice a day, 2 feeds are schedules for the day already, end use case. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful feeding. | Fish are successfully fed. |
| Alt: No food available. | According to estimated food amount, there is no more fish food. |
| Alt: Have been fed in last 2 hours. | Gap between feedings must be greater than 2 hours. |
| Alt: Have been fed twice during the day. | Fish cannot be fed more than twice during each day. |
| Alt: Maximum amount of feeds are already scheduled for the day. | Fish cannot be fed more than twice during each day; two feeds are already scheduled. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful feeding. | Adequate feed in feeder, daily limit not reached, gap between feedings over 2 hours. | Fish are fed, food amount decreased, daily limit usage is increased. Time of feeding logged. |
|  | Alt: No food available. | Inadequate amount of food in feeder, daily limit not reached, gap between feedings over 2 hours. | Fish are not fed, food amount not decreased, daily limit usage is not increased.  Error message: “No food in feeder”; options displayed. |
|  | Alt: Have been fed in last 2 hours. | Adequate amount of food, daily limit not reached, fish have been fed within the last 2 hours | Fish are not fed, food amount not decreased, daily limit usage is not increased.  Error message: “Fish have been fed recently”; options displayed. |
|  | Alt: Have been fed twice during the day. | Adequate amount of food in feeder, daily limit has been reached, gap between feedings over 2 hours. | Fish are not fed, food amount not decreased, daily limit usage is not increased.  Error message: “Cannot feed more than twice in a day”; options displayed. |
|  | Alt: Maximum amount of feeds are already scheduled for the day. | Adequate amount of food in feeder, daily limit has been reached, gap between feedings over 2 hours. | Fish are not fed, food amount not decreased, daily limit usage not increased. Error message: “Two feeds are already scheduled for the day:’ options displayed. |

#### Use Case Name: Automatic Timed feeding

**Actor(s)**: User

**Use Case Precondition(s)**:

* A valid schedule must be made
* Automatic feeding must be enabled

**Use Case Successful post condition(s)**:

* Fish are fed automatically

**Applicable Business Rules:**

* Manual feeding is disabled if 2 scheduled feeds are scheduled for the day
* Manual feeding is disabled 2 hours before and after a scheduled feed

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *System timer* | System (App) |
| 1 | Request to feed the fish. | System checks if any food is available, if food available, feeder is activated. Feeder is deactivated after desired amount of food is fed. Feeder is not yet empty. Time of feeding logged. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | No food available. | Notify user that there is no more food in the feeder, end use case. |
| A2 | Food runs out after feeding. | Fish are fed, notification is sent to user about no food in feeder. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful feeding. | Fish are successfully fed. |
| Alt: No food available. | According to estimated food amount, there is no more fish food. |
| Alt: All food has been finished after feeding. | Fish are fed, feeder needs to be refilled. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful feeding. | Adequate food in feeder, daily limit not reached, and feeder is not empty. | Fish are fed, food amount decreased. Time of feeding logged. |
|  | Alt: No food available. | Inadequate amount of food in feeder. | Fish are not fed, food amount not decreased,  Error message: “No food in feeder”; options displayed. |
|  | Alt: Food has just been finished. | Adequate amount of food upon feeding, feeder is empty after feeding. | Fish are fed, food amount decreased, feeder empty.  Notification message: “Feeder food is empty, please refill”. |

### Tank

A sub-system that enables users to manage and set-up info for each individual tank. The Pi-Seas system is designed to control multiple tanks simultaneously. Once a tank is added, the system will allow users to choose default settings as well as allow customization through editing settings relating to other sub-system use cases.

#### Use Case Name: Add Tank

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Add tank” mobile app page

**Use Case Successful post condition(s)**:

* Tank is added to lank list in app

**Applicable Business Rules:**

* Must not connect to the same tank twice
* Tank code must be valid
* Must have unique tank name

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Enters tank code and password. | Authenticates code and password with connection to server, displays tank information page. |
| 2 | User enters tank information and saves. | Checks for unique tank name, if is unique then saves tank information to PiSeas Tank System and prompts user to visit tank settings pages to review settings. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Code and password authentication failed. | Notify users and go to step 1 of main flow. |
| A2 | Leaves the page without saving. | Does not save any information, end use case. |
| A3 | User does not enter tank name and saves. | Prompt user to enter unique tank name. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Tank added successfully. | User added tank to the tank list of the app. |
| Alt: Failed authentication. | User does not enter correct code and password that matches. |
| Alt: User leaves before saving. | User does not click “save tank information” button and leaves the page. |
| Alt: No tank name entered. | User pressed save without entering tank name. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Tank added successfully | Tank code and passwords are correct and all information has been filled in. | Tank is added to the tank list on the app, all information is saved in the PiSeas System. |
|  | Main: Tank added successfully. | Tank code and passwords are correct and only tank name has been filled in, the rest are set to default. | Tank is added to the tank list on the app, tank name is saved in the PiSeas System. |
|  | Alt: Failed authentication. | Tank code and passwords do not match. | Error message: “incorrect password or code”, let user try again. |
|  | Alt: User leaves before saving. | Tank code and passwords are correct, it doesn’t matter what information is entered, user did not select save button. | The appropriate screen is displayed. |
|  | Alt: No tank name entered. | Tank code and passwords are correct, but no tank name is entered when user pressed save | Error message: “You must enter tank name”, let user try again |

#### Use Case Name: Remove Tank

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Tank List” mobile app page
* Must have at least one tank connected

**Use Case Successful post condition(s)**:

* Tank is disconnected from app

**Applicable Business Rules:**

* Data saved on the phone about disconnected tank will be removed once tank is disconnected.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Selects tank from list. | Displays tank’s detailed info page. |
| 2 | Selects remove tank. | Requests user to confirm deletion, inform user that tank will revert to default automatic settings. |
| 3 | User confirms deletion. | Disconnects tank from user’s app, and displays confirmation message. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Cancels or does not confirm. | Returns to display of tank’s detailed info page and does not disconnect tank. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful Removal. | Tank is disconnected from app, info is deleted and they tank system is set to default settings. |
| Alt: User Cancels. | Tank is not removed from app. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful Removal. | Must have users confirm. | Tank is disconnected, data is deleted and settings return to default. |
|  | Alt: User Cancels. | User does not click confirm. | Nothing changes. |

#### Use Case Name: Update Tank Info

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Tank List” mobile app page
* Must have at least one tank connected

**Use Case Successful post condition(s)**:

* Tank info is updated and saved

**Applicable Business Rules:**

* Must have unique tank name

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Selects tank from list | Displays tank’s detailed info page. |
| 2 | Enters new tank info and saves. | Checks for unique tank name, if it is unique then saves tank information to PiSeas Tank System. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Leaves the page without saving. | Does not save any information, end use case. |
| A2 | User does not enter valid tank name. | Prompt user to enter unique tank name. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful Update. | Tank info is updated both on app and the PiSeas Tank System. |
| Alt: User Cancels. | Tank info is not changed. |
| Alt: No Valid Tank Name. | User did not enter valid or unique tank name, prompt them to try again. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful Update. | Valid and unique tank name. | Saves info in app and PiSeas Tank System. |
|  | Alt: User Cancels. | User does not click save. | Nothing changes. |
|  | Alt: No Valid Tank Name. | Invalid or non-unique tank name. | Prompt user to enter tank name then press save. |

#### Use Case Name: Checking Notifications

**Actor(s)**: Users

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must have at least one tank connected

**Use Case Successful post condition(s)**:

* User sees all notifications linked to tank

**Applicable Business Rules:**

* There is a max number of pending notifications. If this is reached, an alarm will sound on the PiSeas Tank System
* If user selects a push notification, then it will lead them straight to step 2
* Once a notification is seen, it is no longer pending on the system

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Selects tank from list. | Displays tank’s detailed info page. |
| 2 | Selects notifications. | Displays all pending notifications as well as history of older notifications. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful Checking of Notifications. | User has seen all notification and system will clear pending notifications. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful Checking of Notifications. | Checks for pending notifications. | Displays notifications and system will change status of notifications from pending to history. |

#### Use Case Name: Updating App Settings

**Actor(s)**: User

**Use Case Precondition(s)**:

* Must be in the “Settings” mobile app page

**Use Case Successful post condition(s)**:

* Saves user`s preferences to app settings.

**Applicable Business Rules:**

* By default, the app does not have a favourite tank and will provide push notifications.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Selects on/off for push notifications and a tank from tank list drop down for favorite tank and save. | Displays settings page with updated settings and a confirmation message that says new settings are saved. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful Settings Update | New settings are saved on the app. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful Settings Update | User clicked save | Saves new settings on app. |

### Water

Water analysis tools include a pH sensor for testing the pH levels and a conductivity sensor for measuring the salinity of the water. If enabled the system will automatically provide analysis and notify user when water quality drops.

#### Use Case Name: Enable automatic Water Flow

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the water flow management page
* Automatic water flow is disabled

**Use Case Successful post condition(s)**:

* Automatic water flow is activated

**Applicable Business Rules:**

* Both high and low water sensors must be connected
* Power relay must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | User enables automatic water flow. | System will manage water levels. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Automatic water flow is unable to turn on due to missing water level sensor(s) and/or power relay. | User is redirected to the “water flow management” page while notifying the user of the missing sensor(s) or power relay. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic water flow turned on successfully. | The system will automatically manage the water levels of the fish tank. |
| Alt: System is unable to turn on automatic water flow due to missing water level sensor(s) and/or power relay. | The system is missing a sensor and/or a power relay to automatically control water levels of the tank. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic water flow turned on successfully. | Water level sensor(s) and power relay are connected to the system. | System is automatically regulating water levels of the tank. |
|  | Alt: System is unable to turn on automatic water flow due to missing water level sensor(s). | Water level sensor(s) and power relay are not connected to the system. | System notifies the user via prompt or message specifying which specific sensor(s) and/or power relay need to be attached to the system. |

#### Use Case Name: Disable automatic Water Flow

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the water flow management page
* Automatic water flow is enabled

**Use Case Successful post condition(s)**:

* Automatic water flow is deactivated

**Applicable Business Rules:**

* Power relay must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | User disables automatic water flow. | System will no longer manage water levels. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic water regulation turned off. | The system will not manage the tank’s water level automatically. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic water regulation turned off. | Power relay is connected to the system. | System not automatically regulating the water levels of the tank. |

#### Use Case Name: Activate Water Output Pump

**Actor(s)**: User, System timer

**Use Case Precondition(s)**:

* User must be on the water flow management page
* Tank is not empty
* Water output pump not already active

**Use Case Successful post condition(s)**:

* Water Output pump is turned on

**Applicable Business Rules:**

* Both water level sensors must be connected
* Power relay device must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): | System |
| 1 | Requests to activate water output pump. | System turns on water output pump. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | No enough water in tank. | System does not turn on water output flow. |
| A2 | Pump is unable to activate. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Water output pump activated. | The system turns on the water output pump. |
| Alt: The tank has not enough water in the tank. | The tank does not have enough water for the water output to function properly. |
| Alt: System is unable to turn on the water output pump due to missing water level sensor(s) and/or power relay. | The system is missing a sensor and/or a power relay to activate the water output pump. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Water output pump activated. | Water level sensor(s) and power relay are connected to the system. | Water output turned on successfully. |
|  | Alt: The tank has not enough water in the tank. | Low water level sensor does not detect enough water. | System notifies the user via prompt or message stating that the water levels in the tank is too low. |
|  | Alt: System is unable to deactivate the water output pump due to missing water level sensor(s). | Water level sensor(s) and power relay are connected to the system. | System notifies the user via prompt or message specifying which specific sensor(s) and/or power relay need to be attached to the system. |

#### Use Case Name: Deactivate Water Output Pump

**Actor(s)**: User, System timer

**Use Case Precondition(s)**:

* Water output pump activated
* User is on the water flow management page

**Use Case Successful post condition(s)**:

* Water output pump is deactivated

**Applicable Business Rules:**

* Power relay device must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User, System timer* | System |
| 1 | Requests to deactivate water output pump. | System turns off water output pump. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Pump is unable to turn off due to missing power relay device. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Water output pump is deactivated. | System deactivates the water output pump. |
| Alt: Pump is unable to deactivate due to missing power relay device. | The system is unable to detect the power relay device to deactivate water output pump. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Water output pump is deactivated. | Power relay is connected to the system. | System deactivates the water output pump. |
|  | Alt: System is unable to turn on automatic water flow due to missing water level sensor(s) and/or power relay. | Power relay is connected to the system. | System notifies the user via prompt or message that the power relay needs to be attached to the system. |

#### Use Case Name: Activate Water Input Pump

**Actor(s)**: User, System timer

**Use Case Precondition(s)**:

* User must be on the water flow management page
* Tank is not empty
* Water input pump not already active

**Use Case Successful post condition(s)**:

* Water input pump is activated

**Applicable Business Rules:**

* Both water level sensors must be connected
* Power relay device must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): | System |
| 1 | Requests to activate water input pump. | System turns on water input pump. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Too much water in the tank. | System does not turn on water input flow. |
| A2 | Pump is unable to activate. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Water input pump turned on. | The system turns on the water input pump. |
| Alt: Too much water in tank. | Top water level sensor is activated, indicating there is too much water. |
| Alt: System is unable to turn on the water input pump due to missing water level sensor(s) and/or power relay. | The system is missing a sensor and/or a power relay to activate the water input pump. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Water input pump turned on. | Water level sensor(s) and power relay are connected to the system. | Water input turned on successfully. |
|  | Alt: Too much water in the tank | High water level sensor is active. | System notifies the user via prompt or message stating that the water levels in the tank is too high. |
|  | Alt: System is unable to turn on the water input pump due to missing water level sensor(s). | Water level sensor(s) and power relay are connected to the system. | System notifies the user via prompt or message specifying which specific sensor(s) and/or power relay need to be attached to the system. |

#### Use Case Name: Deactivate Water Input Pump

**Actor(s)**: User, System timer

**Use Case Precondition(s)**:

* Water input pump activated
* User is on the water flow management page

**Use Case Successful post condition(s)**:

* Water input pump is deactivated

**Applicable Business Rules:**

* Power relay device must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User, System timer* | System |
| 1 | Requests to deactivate water input pump. | System turns off water input pump. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Pump is unable to deactivate due to missing power relay device. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Water input pump is deactivated. | System deactivates the water input pump. |
| Alt: Pump is unable to deactivate due to missing power relay device. | The system is unable to detect the power relay device to deactivate water input pump. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Water input pump is deactivated. | Power relay is connected to the system. | System deactivates the water input pump. |
|  | Alt: System is unable to turn on automatic water flow due to missing water level sensor(s) and/or power relay. | Power relay is connected to the system. | System notifies the user via prompt or message that the power relay needs to be attached to the system. |

#### Use Case Name: Check water level

**Actor(s)**: System timer

**Use Case Precondition(s)**:

* Input and/or output water pump must be activated

**Use Case Successful post condition(s)**:

* Water levels checked

**Applicable Business Rules:**

* Both water level sensors must be connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *System timer* | System |
| 1 | System timer requests for the tank’s water level. | System checks the current water level in the tank. Activates water pumps to adjust if water level is not optimal. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| 1 | Sensor(s) are not responding or are no longer connected. | Water flow sub system is temporarily disabled to prevent potential overflow as the water level cannot be checked. System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: System timer is able to check water levels. | System timer requests to the system to check water level. |
| Alt: Sensor(s) are not responding or are no longer connected. | System is unable to check water levels. The system will notify the user and disable any pumps to negate any damages. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: System timer is able to check water levels. | Water level sensor(s) are connected. | Water levels are logged. |
|  | Alt: Sensor(s) are not responding or are no longer connected. | Water level sensor(s) are connected. | System notifies the user via notifications that the water level sensors are not working properly or are missing. The system will disable both water input and output pumps. |

#### Use Case Name: Enable Automatic Water Conductivity Analysis

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the water analysis management page
* Automatic water conductivity is disabled

**Use Case Successful post condition(s)**:

* Automatic water conductivity analysis is turned on

**Applicable Business Rules:**

* Water conductivity sensor is connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to enable the automatic water conductivity analysis. | System enables automatic water conductivity analysis. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Water conductivity threshold values are invalid. | User is asked to set valid acceptable ranges for water conductivity. |
| A2 | Water conductivity sensor is not connected to the system. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic water conductivity analysis turned on. | The system turns on automatic water conductivity analysis. |
| Alt: Water conductivity threshold values are invalid. | The user has inputted invalid values for the water conductivity threshold |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic water conductivity analysis turned on | Water conductivity threshold. Water conductivity sensor connected. | System turns on automatic water conductivity analysis |
|  | Alt: Water conductivity threshold values are invalid. | Water conductivity threshold values are out of reasonable ranges (0-60000). Water conductivity threshold min is higher than threshold max. | System notifies the user via prompt or message specifying that the values current assigned are invalid. |
|  | Alt: System is unable to turn on the automatic water conductivity analysis due to missing water conductivity sensors. | Water conductivity sensor connected. | System notifies the user via prompt or message specifying the water conductivity sensor needs to be attached to the system. |

#### Use Case Name: Enable Automatic pH analysis

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the water analysis management page
* Automatic pH is disabled

**Use Case Successful post condition(s)**:

* Automatic pH analysis is turned on

**Applicable Business Rules:**

* pH sensor is connected

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to enable the automatic pH analysis. | System enables automatic pH analysis. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | pH threshold values are invalid. | User is asked to set valid acceptable ranges for pH. |
| A2 | pH sensor is not connected to the system. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic pH analysis turned on. | The system turns on automatic pH analysis. |
| Alt: pH threshold values are invalid. | The user has inputted invalid values for the pH threshold. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic pH analysis turned on. | pH threshold. pH sensor connected. | System turns on automatic pH analysis. |
|  | Alt: pH threshold values are invalid. | pH threshold values are out of reasonable ranges (0-14). pH threshold min is higher than threshold max. | System notifies the user via prompt or message specifying that the values current assigned are invalid. |
|  | Alt: System is unable to turn on the automatic pH analysis due to missing pH sensors. | Checks if pH sensor connected | System notifies the user via prompt or message specifying the pH sensor needs to be attached to the system. |

#### Use Case Name: Disable Automatic Water Conductivity Analysis

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is off the water analysis management page
* Automatic water conductivity is enabled

**Use Case Successful post condition(s)**:

* Automatic water conductivity analysis is turned off

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to disable the automatic water conductivity analysis. | System disable automatic water conductivity analysis. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic water conductivity analysis turned off. | The system turns off automatic water conductivity analysis. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic water conductivity analysis turned off. | Water conductivity threshold. Water conductivity sensor connected. | System turns off automatic water conductivity analysis. |

#### Use Case Name: Disable Automatic pH analysis

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is off the water analysis management page
* Automatic pH is enabled

**Use Case Successful post condition(s)**:

* Automatic pH analysis is turned off

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to disable the automatic pH analysis. | System disable automatic pH analysis. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic pH analysis turned off. | The system turns off automatic pH analysis. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic pH analysis turned off. | pH threshold. pH sensor connected. | System turns off automatic pH analysis. |

#### Use Case Name: Update water conductivity threshold

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the water conductivity management page

**Use Case Successful post condition(s)**:

* Water conductivity threshold values saved

**Applicable Business Rules:**

* Minimal threshold value cannot be higher than the max

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to update water conductivity threshold values. | System saves the water conductivity threshold values. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Water conductivity threshold values are invalid. | User is asked to set valid acceptable ranges for Water conductivity. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Water conductivity threshold values updated. | System saves water conductivity threshold values. |
| Alt: Water conductivity threshold values are invalid. | The user has inputted invalid values for the water conductivity threshold |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Water conductivity threshold values updated. | Water conductivity threshold values are out of reasonable ranges (0-60000). Water conductivity threshold min is higher than threshold max. | System saves the update values. |
|  | Alt: Water conductivity threshold values are invalid. | Water conductivity threshold values are out of reasonable ranges (0-60000). Water conductivity threshold min is higher than threshold max. | System notifies the user via prompt or message specifying that the values current assigned are invalid. |

#### Use Case Name: Update pH threshold

**Actor(s)**: User

**Use Case Precondition(s)**:

* User is on the pH management page

**Use Case Successful post condition(s)**:

* pH threshold values saved

**Applicable Business Rules:**

* Minimal threshold value cannot be higher than the max

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System |
| 1 | Request to update pH threshold values. | System saves the pH threshold values. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | pH threshold values are invalid. | User is asked to set valid acceptable ranges for pH. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: pH threshold values updated. | System saves pH threshold values. |
| Alt: pH threshold values are invalid. | The user has inputted invalid values for the pH threshold. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: pH threshold values updated. | pH threshold values are out of reasonable ranges (0-14). pH threshold min is higher than threshold max. | System saves the update values. |
|  | Alt: pH threshold values are invalid. | pH threshold values are out of reasonable ranges (0-14). pH threshold min is higher than threshold max. | System notifies the user via prompt or message specifying that the values current assigned are invalid. |

#### Use Case Name: Check pH Levels

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* pH Sensor is connected

**Use Case Successful post condition(s)**:

* pH levels readings are saved on the PiSeas Tank System

**Applicable Business Rules:**

* System will automatically check pH values every min
* Notification will be issued if pH is not within the acceptable range

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Requests pH reading. | System checks pH readings, checks them against acceptable range and saves them. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If pH is not within acceptable range. | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful pH check. | pH value was checked and in range. |
| Alt: pH is out of range. | pH value was checked and out of range. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful pH check. | pH value was checked and in range. | System saves pH reading for future display. |
|  | Alt: pH is out of range. | pH value was checked and out of range. | System saves pH reading and sends notification to alert user. |

#### Use Case Name: Check Water Conductivity Levels

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Conductivity Sensor is connected

**Use Case Successful post condition(s)**:

* Conductivity levels readings are saved on the PiSeas Tank System

**Applicable Business Rules:**

* System will automatically check conductivity values every min
* Notification will be issued if conductivity is not within the acceptable range

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Requests conductivity reading | System checks conductivity readings, checks them against acceptable range and saves them |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If conductivity is not within acceptable range | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful conductivity check | Conductivity value was checked and in range. |
| Alt: Conductivity is out of range | Conductivity value was checked and out of range |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful pH check | Conductivity value was checked and in range. | System saves conductivity reading for future display |
|  | Alt: Conductivity is out of range | Conductivity value was checked and out of range | System saves pH reading and sends notification to alert user |

### Lighting Sub-System

Lamps are used to provide proper lighting to the plants and animals living inside the tank. Lights will simulate sun shine during the day and turn off at night. The lights will be set to turn on and off during sunset and sunrise times, to make the lighting as natural as possible. The app will also let the user interact with the aquarium by letting the user manually turn the lights on and off. The system will disable the manual controls if the users turns the lights on/off too many times.

#### Use Case Name: Create Light Schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* Must be on the “Add Light Schedule” activity.

**Use Case Successful post condition(s)**:

* Light schedule is created and stored.

**Applicable Business Rules:**

* Cannot create a light schedule with repeating times.
* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User selects a time and the light state (either on or off). | System displays time and light state. Requests confirmation from the user. |
| 2 | User confirms the settings. | System saves light settings then redirects to the “update light schedule” page. |
| 3 | Repeat steps 1-2 until user selects to go back to the “tank management” page. | System displays the light schedule and redirects to the “tank management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User leaves page before saving light schedule. | System does not save schedule. |
| A2 | User cancels. | System does not save the schedule. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Light schedule created. | The light schedule has been successfully created and stored. |
| Alt: Light schedule not stored. | The light schedule was not stored because the time is already stored. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Light schedule created. | Valid data entered (time and state). | Light schedule created and stored. User redirected to “Manage Light” activity. |
|  | Alt: Light schedule not stored. | The system is unable to store the schedule because the time is already stored. | Error “Unable to create light schedule”. |

#### Use Case Name: Update Light Schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* Must be on the “Update Light Schedule” activity.

**Use Case Successful post condition(s)**:

* Light schedule is created and stored.

**Applicable Business Rules:**

* Cannot have two light schedule with overlapping times.
* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User selects a light time to modify. | System displays the light time and state. Requests the new light time and state. |
| 2 | User enters a time and a light state. | System displays light time and the state. Requests confirmation from the user. |
| 3 | User confirms the settings. | System saves the light settings then redirects to the “update light schedule” page. |
| 4 | Repeat steps 1-3 until the user decides to go back to the “tank management” page. | System displays the light schedule and redirects to the “tank management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User leaves the page before saving update. | System does not save the new schedule. |
| A2 | User cancels. | System does not save the settings and redirects to the “update light” page. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Light schedule updated and stored. | The light schedule is updated and stored on the server. |
| Alt: Light schedule not updated due to the time being already stored. | The light schedule was not stored due to the selected light time already being stored in the schedule. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Light schedule updated. | Valid data entered (time and state). | Light schedule updated and stored. User redirected to “Manage Light” activity. |
|  | Alt: Light schedule not updated. | Repeating time has been entered. | Error “Cannot update light schedule due to repeating time”. |

#### Use Case Name: Delete Light Schedule

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* Must be on the “Update Light Schedule” activity.
* There must be a light schedule stored.

**Use Case Successful post condition(s)**:

* Light schedule is deleted.

**Applicable Business Rules:**

* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User selects light time to remove. | System displays light time and light state. The system then requests for confirmation on the removal of the displayed light time. |
| 2 | User confirms removal. | System removes the light settings, displays the removed settings and redirects to the remove “light schedule” page. |
| 3 | Repeat steps 1-2 until the user decides to go back to the “tank management” page. | System displays the current light schedule and redirects to the “tank management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User leaves the page. | System does not save schedule. |
| A2 | User cancels the operation. | System does not save the settings, and redirects to the “light schedule” page. |
| A3 | There are no more light times. | System displays a message stating there are no light times stored. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Light schedule deleted. | The light schedule has been deleted from the system. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Light schedule deleted. | User confirmation reply. | Light schedule is deleted and the user is redirected to the “Manage Light” activity. |

#### Use Case Name: Enable Automated Lighting

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Light Management” activity.

**Use Case Successful post condition(s)**:

* Automated Lighting is enabled.

**Applicable Business Rules:**

* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User enables automatic light setting. | System displays the current light schedule and requests confirmation. |
| 2 | User confirms. | System saves the changes, displays the light schedule and redirects to the “light management" page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User cancels. | System does not save changes and redirects to the “light management” page. |
| A2 | User leaves page. | System does not save any changes. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic lighting enabled. | The system enables automatic lighting regulation successfully. |
| Alt: Unable to enable automatic lighting. | The system is unable to enable automatic lighting due to no schedule being stored. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic lighting enabled. | Valid lighting schedule stored. | System enables automatic lighting. |
|  | Alt: Unable to enable automatic lighting. | There is no schedule stored in the system. | User is redirected to the “Update Light Schedule” activity; show options. |

#### Use Case Name: Disable Automatic Lighting

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Light Management” activity.

**Use Case Successful post condition(s)**:

* Automated Lighting is disabled.

**Applicable Business Rules:**

* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User disables automatic light setting. | System displays the current light schedule and requests confirmation. |
| 2 | User confirms. | System saves the changes, displays the current light state and redirects to the “light management" page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User cancels. | System does not save changes and redirects to the “light management” page. |
| A2 | User leaves page. | System does not save any changes. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic lighting disabled. | Automatic lighting is disabled. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic lighting disabled. | User confirmation reply. | System disables automatic lighting. |

#### Use Case Name: Turn On Lights Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Light Management” activity.
* Light must be connected to the tank.

**Use Case Successful post condition(s)**:

* Lights turn on.

**Applicable Business Rules:**

* Manually turning lights on disables automatic lighting.
* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn on lights for the tank. | System turns the lights on, displays a confirmation message. |
| 2 | User confirms | System redirects to the “Light Management” activity. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User does not accept confirmations. | System goes back to the “light management” page, does not turn lights on. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Lights turn on. | The system turns on the lights successfully. |
| Alt: Lights stay off. | The system does not turn on the lights. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Lights turn on. | User confirmation reply. | The system turns on the lights. Automatic lighting is disabled. |
|  | Alt: Lights stay off. | User does not confirm reply. | The system leaves the lights turned off. |

#### Use Case Name: Turn Off Lights Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Light Management” activity.
* Light must be connected to the tank.

**Use Case Successful post condition(s)**:

* Lights turn off.

**Applicable Business Rules:**

* Manually turning lights off disables automatic lighting.
* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn off lights for the tank. | System turns the lights off, displays a confirmation message. |
| 2 | User confirms | System redirects to the “light management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | User does not accept confirmations. | System goes back to the “light management” page, does not turn lights off. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Lights turn off. | The system turns off the lights successfully. |
| Alt: Lights stay on. | User does not accept confirmation. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Lights turn off. | User confirmation reply. | The system turns off the lights. |
|  | Alt: Lights stay on. | User does not confirm reply. | The system leaves the lights turned on. |

#### Use Case Name: System Timer Turns On Lights

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Light schedule is set.
* Automatic lighting is enabled.
* Light is connected.

**Use Case Successful post condition(s)**:

* Light turns on.

**Applicable Business Rules:**

* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | System Timer requests to turn on light based on the light schedule. | System turns on the light. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Lights are unable to turn on. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Scheduled light on time. | The light schedule has the current time stored, and the light is supposed to turn on. |
| Alt: The current time is not in the light schedule. | The current time is not in the light schedule. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: It is time to turn on the light. | Compare the current time to the times stored in the light schedule. | The lights turn on. |
|  | Alt: The current time is not in the light schedule. | The system compares the current time to the times stored in the light schedule. | The system continues comparing the current time to the times in the light schedule. |

#### Use Case Name: System Timer Turns Off Lights

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Light schedule is set.
* Automatic lighting is enabled.
* Light is connected.

**Use Case Successful post condition(s)**:

* Light turns off.

**Applicable Business Rules:**

* System unable to check automatically whether light is in working condition, users must check themselves.

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | System Timer requests to turn on light based on schedule. | System turns off the lights. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Lights are unable to turn off. | System alerts user on their smartphone device through notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: It is time to turn off the light. | The light schedule has the current time stored, and the light is supposed to turn off. |
| Alt: The current time is not in the light schedule. | The current time is not in the light schedule. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: It is time to turn off the light. | Compare the current time to the times stored in the light schedule. | The lights turn off. |
|  | Alt: The current time is not in the light schedule. | The system compares the current time to the times stored in the light schedule. | The system continues comparing the current time to the times in the light schedule. |

### Temperature Sub-System

The system will adjust the temperature of the water through the use of the temperature sensors, a heater and a fan. The system will ensure the temperature of the environment remains in acceptable ranges and will alert the user if it cannot self-adjust in time. The settings for the acceptable ranges will be accessible through the mobile app interface.

#### Use Case Name: Add Temperature Range

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Temperature manager” mobile app page
* Temperature range must not exist

**Use Case Successful post condition(s)**:

* Temperature range is added

**Applicable Business Rules:**

* Only 1 temperature range can be set
* A minimum and maximum value must be selected
* Maximum value must be greater than the minimum value

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): *User* | System (App) |
| 1 | Select “add temperature range” | System opens page for adding temperature, displaying a general range (minimum: 25°, maximum: 27°C) |
| 2 | Select minimum and maximum values for the range | System checks if values are acceptable. If values are acceptable, maximum value is greater than the minimum, system asks user for confirmation. Acceptable range is between 10°C and 40°C |
| 3 | Confirm new temperature range | System saves the temperature range and returns to the “temperature main” page |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Maximum value too high | Display error message, display highest temperature possible |
| A2 | Minimum value too low | Display error message, display lowest temperature possible |
| A3 | Minimum value greater than maximum value | Display error message, ask user to re-enter |
| A4 | User cancels | No information is saved, go back to “temperature main” screen |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Correct temperature set | Valid temperature selected, temperature range saved |
| Alt: Invalid maximum temperature | Maximum temperature range too high |
| Alt: Invalid minimum temperature | Minimum temperature range too low |
| Alt: Invalid temperatures | Minimum temperature is greater than the maximum |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Correct temperature set | Maximum and minimum range set, maximum and minimum values are within range, maximum is greater than minimum | Temperature range is set, screen is redirected to the temperature main page, maximum and minimum values in the temperature main page are updated |
|  | Alt: Invalid maximum temperature | Maximum and minimum range set, maximum range too high, minimum range valid, maximum greater than minimum value | Temperature range not saved, page shows maximum acceptable temperature.  Error message: “Maximum temperature too high”; options displayed |
|  | Alt: Invalid minimum temperature | Maximum and minimum range set, minimum range too low, Maximum range valid, maximum greater than minimum value | Temperature range not saved, page shows minimum acceptable temperature.  Error message: “Minimum temperature too low”; options displayed |
|  | Alt: Invalid temperatures | Maximum and minimum range set, maximum and minimum values are within range, minimum is greater than maximum | Temperature range not saved. Error message: “Minimum temperature must be lower than the maximum temperature”; options displayed |

#### Use Case Name: Update Temperature Range

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server
* Must be on the “Temperature manager” mobile app page
* Temperature range must already exist

**Use Case Successful post condition(s)**:

* Temperature range is updated

**Applicable Business Rules:**

* Only 1 temperature range can be set
* A minimum and maximum value must be selected
* Maximum value must be greater than the minimum value

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Select “update temperature range” | System opens page for adding temperature, display previous maximum and minimum values |
| 2 | Select minimum and maximum values for the range | System checks if values are acceptable. If values are acceptable, maximum value is greater than minimum, system asks user for confirmation |
| 3 | Confirm update | System saves the temperature range and returns to the “temperature main” page |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Maximum value too high | Display error message, display highest temperature possible |
| A2 | Minimum value too low | Display error message, display lowest temperature possible |
| A3 | Minimum value greater than maximum value | Display error message, ask user to re-enter |
| A4 | User cancels | Previous temperature not changed, go back to “temperature main” screen without saved changes |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Correct temperature set | Valid temperature selected, temperature range saved |
| Alt: Invalid maximum temperature | Maximum temperature range too high |
| Alt: Invalid minimum temperature | Minimum temperature range too low |
| Alt: Invalid temperatures | Minimum temperature is greater than the maximum |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Correct temperature set | Maximum and minimum range set, maximum and minimum values are within range, maximum is greater than minimum | Temperature range is set, screen is redirected to the temperature main page, maximum and minimum values in the temperature main page are updated |
|  | Alt: Invalid maximum temperature | Maximum and minimum range set, maximum range too high, minimum range valid, maximum greater than minimum value | Temperature range not saved, page shows maximum acceptable temperature.  Error message: “Maximum temperature too high”; options displayed |
|  | Alt: Invalid minimum temperature | Maximum and minimum range set, minimum range too low, Maximum range valid, maximum greater than minimum value | Temperature range not saved, page shows minimum acceptable temperature.  Error message: “Minimum temperature too low”; options displayed |
|  | Alt: Invalid temperatures | Maximum and minimum range set, maximum and minimum values are within range, minimum is greater than maximum | Temperature range not saved. Error message: “Minimum temperature must be lower than the maximum temperature”; options displayed |

#### Use Case Name: Remove Temperature Range

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server
* Must be on the “Temperature manager” mobile app page
* Temperature range must already exist

**Use Case Successful post condition(s)**:

* Temperature range is removed

**Applicable Business Rules:**

* If Automatic Temperature is enabled, removing range will disable the automatic temperature regulation

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Select “delete temperature range” | System opens confirms with user whether they want to remove the temperature range |
| 2 | User confirms the removal | System checks if automatic temperature regulation is enabled, system disables the automatic temperature regulation. The temperature range is removed from the system |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Automatic temperature range disable | Automatic temperature range is left disabled; temperature range is removed from the system |
| A2 | User cancels removal | System goes back to the “temperature main” page, temperature range is not removed |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Temperature range is removed, automatic regulation enabled | System successfully deletes the temperature range; automatic regulation is disabled |
| Alt: Temperature range is removed, automatic regulation disabled | System successfully deletes the temperature range |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Temperature range is removed, automatic regulation enabled | Automatic regulation is enabled | Temperature range is deleted, automatic temperature regulation is disabled, page is redirected to the “temperature main” page |
|  | Alt: Temperature range is removed, automatic regulation disabled | Automatic regulation is disabled | Temperature range is deleted; page is redirected to the “temperature main” page |

#### Use Case Name: Turn Heater On Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Temperature Management” page.

**Use Case Successful post condition(s)**:

* Heater is turned on.

**Applicable Business Rules:**

* Manually turning on the heater disables temperature automation

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn on heater for the tank. | System turns the heater on, displays a confirmation message. Checks current temperature against temperature in 10mins, if temperature increases, end use case. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If in 10mins the temperature does not increase | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful turn on | Heater has been successfully turn on and is now affecting water temperature |
| Alt: No temp change | Heater has been set to turn on but is not affecting water temperature |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful turn on | Checks and compares two temperature readings, temperature has increased | Heater was turned on successfully, disable temperature automation feature |
|  | Alt: No temp change | Checks and compares two temperature readings, no change | Heater was not effective, alert user and disable temperature automation feature |

#### Use Case Name: Turn Heater Off Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Temperature Management” page.

**Use Case Successful post condition(s)**:

* Heater is turned off.

**Applicable Business Rules:**

* Manually turning off the heater disables temperature automation

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn off heater for the tank. | System turns the heater off, displays a confirmation message. Checks current temperature against temperature in 10mins, if temperature does not continue to increase, end use case. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If in 10mins the temperature continues to increase | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful turn off | Heater has been successfully turn off and is now affecting water temperature |
| Alt: Temperature continues to increase | Heater has been set to turn off but is still affecting water temperature |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful turn off | Checks and compares two temperature readings, temperature has stopped increasing | Heater was turned off successfully, disable temperature automation feature |
|  | Alt: Temperature continues to increase | Checks and compares two temperature readings, temperature continues to increase | Heater was not affected, alert user and disable temperature automation feature |

#### Use Case Name: Turn Fan On Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Temperature Management” page.

**Use Case Successful post condition(s)**:

* Fan is turned on.

**Applicable Business Rules:**

* Manually turning on the fan disables temperature automation

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn on fan for the tank. | System turns the fan on, displays a confirmation message. Checks current temperature against temperature in 10mins, if temperature decreases, end use case. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If in 10mins the temperature does not decrease | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful turn on | Fan has been successfully turn on and is now affecting water temperature |
| Alt: No temp change | Fan has been set to turn on but is not affecting water temperature |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful turn on | Checks and compares two temperature readings, temperature has decreased | Fan was turned on successfully, disable temperature automation feature |
|  | Alt: No temp change | Checks and compares two temperature readings, no change | Fan was not effective, alert user and disable temperature automation feature |

#### Use Case Name: Turn Fan Off Manually

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the server.
* User must be on the “Temperature Management” page.

**Use Case Successful post condition(s)**:

* Fan is turned off.

**Applicable Business Rules:**

* Manually turning off the fan disables temperature automation

**Use Case Specifications**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | Turn off fan for the tank. | System turns the fan off, displays a confirmation message. Checks current temperature against temperature in 10mins. If temperature does not continue to decrease, end use case. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | If in 10mins the temperature continues to decrease | System alerts user on their smartphone device through the notification system. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Successful turn off | Fan has been successfully turn off and is now affecting water temperature |
| Alt: Temperature continues to decrease | Fan has been set to turn off but is still affecting water temperature |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Successful turn off | Checks and compares two temperature readings, temperature has stopped decreasing | Fan was turned off successfully, disable temperature automation feature |
|  | Alt: Temperature continues to decrease | Checks and compares two temperature readings, temperature continues to decrease | Fan was not affected, alert user and disable temperature automation feature |

#### Use Case Name: Enable Automatic Temperature Regulation

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Temperature Management” mobile app page

**Use Case Successful post condition(s)**:

* Automatic temperature regulation is activated

**Applicable Business Rules:**

* Temperature device must be connected and working
* System will check temperature every min

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User enables automatic temperature regulation. | System displays the current temperature range and requests confirmation. |
| 2 | User confirms. | System saves the changes and redirects to the “temperature management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | System is unable to enable automatic temperature regulation. | System sends a notification to the user. System alerts user on their smartphone device through notification system. |
| A2 | User leaves page before confirming. | System does not save any changes. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic temperature regulation turned on successfully | The system will automatically manage the temperature of the fish tank. |
| Alt: System is unable to turn on the automatic temperature regulation due to a unresponsive sensor or device(s) | The system will notify the user of the unresponsive sensor(s). |
| Alt: System is unable to turn on the automatic temperature regulation as the user does not accept changes | The system will not save any changes and will redirect the user back to the main temperature management page. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic temperature regulation turned on successfully | Temperature ranges are valid. Fan, heater and temperature sensors are connected to the system. | System is automatically regulating temperature of the fish tank. |
|  | Alt: System is unable to turn on the automatic temperature regulation due to a missing sensor or device(s) | Fan, heater and temperature sensors are connected to the system. | System notifies the user via prompt or message specifying which specific device(s) or sensor need to be attached to the system. |
|  | Alt: System is unable to turn on the automatic temperature regulation due to invalid temperature range | Temperature ranges given are not too high (above 30 and below 15 degrees) and max temperature range is higher than the minimal. | System notifies the user via prompt or message that the given temperature ranges are invalid. System will ask the user to input the values once again. |
|  | Alt: System is unable to turn on the automatic temperature regulation as the user does not accept changes | Temperature ranges are valid. Fan, heater and temperature sensor is connected to the system. Doesn’t matter since values are not saved. | System will redirect the user to the tank management page. System will not save any changes. |

#### Use Case Name: Disable Automatic Temperature Regulation

**Actor(s)**: User

**Use Case Precondition(s)**:

* Phone must be connected to the internet
* Must be in the “Temperature Management” mobile app page
* Automatic temperature regulation must be enabled

**Use Case Successful post condition(s)**:

* Automatic temperature regulation is disabled

**Applicable Business Rules:**

* Temperature device are connected
* System will check temperature every min

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): User | System |
| 1 | User disables automatic temperature regulation. | System displays the current temperature range and requests confirmation. |
| 2 | User confirms. | System saves the changes and redirects to the “temperature management” page. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | System is unable to disable automatic temperature regulation. | System sends a notification to the user. System alerts user on their smartphone device through notification system. |
| A2 | User leaves page before confirming. | System does not save any changes. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Automatic temperature regulation turned off successfully | The system will not manage the tanks temperature automatically |
| Alt: System is unable to turn off the automatic temperature regulation as the user does not accept changes | The system will not save any changes and will redirect the user back to the main temperature management page. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Automatic temperature regulation turned off successfully | Temperature ranges are valid. Fan, heater and temperature sensors are connected to the system. | System is not regulating temperature of the fish tank. |
|  | Alt: System is unable to turn off the automatic temperature regulation as the user does not accept changes | Temperature ranges are valid. Fan, heater and temperature sensor is connected to the system. Doesn’t matter since values are not saved. | System will redirect the user to the tank management page. System will not save any changes. |

#### Use Case Name: System Timer Turns Heater On

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Temperature range is set.
* Automatic temperature regulation is enabled.
* Temperature sensor it connected to the tank system.
* Tank is filled with water.

**Use Case Successful post condition(s)**:

* Heater turns on and starts warming up the water in the tank.

**Applicable Business Rules:**

* Heater can only be turned on if the temperature is below the minimum temperature

**Use Case Specification**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Request temperature check. | System returns the temperature reading. Temperature is below the min range. System turns the heater on. |
| 2 | Request temperature check, after 5min. | System compares temperature reading to previous reading to see if the heater was turned on, thus affecting water temperature. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Temperature is above the min range. | End use case |
| A2 | System is unable to get temperature reading. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |
| A3 | System is unable to turn on the heater. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Heater turned on. | The system turns the heater on and starts warming the water in the tank. |
| Alt: No temperature reading. | The system is unable to get a temperature reading from the sensor. |
| Alt: Heater does not turn on. | The system attempts to turn the heater on but after 5mins the temperature does not change. |
| Alt: Current temperature (of the water in the tank) is above the minimum. | The temperature is above the minimum. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Heater turned on. | The temperature is below the minimum, and the temperature increases. | The heater is on and the temperature of the water is increasing. System continues monitoring the temperature. |
|  | Alt: No temperature reading. | The system does not get a temperature reading from the sensor. | Error message: “Unable to obtain temperature reading”; the system does not attempt to turn the heater on. Notification created and synced to phone (when connected). |
|  | Alt: Heater does not turn on. | System attempts to turn on the heater and checks the temperature after 5mins for a temperature increase. | Error message: “Heater cannot turn on”; notification created and synced to phone (when connected). |
|  | Alt: Current temperature (of the water in the tank) is above the minimum. | System compares the current temperature to the minimum. | System turns off the heater and continues monitoring the temperature checking if it falls below the minimum. |

#### Use Case Name: System Timer Turns Heater Off

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Temperature range is set.
* Automatic temperature regulation is enabled.
* Temperature sensor is connected to the tank.
* Tank is filled with water.

**Use Case Successful post condition(s)**:

* Heater turns off and stops heating the water in the tank.

**Applicable Business Rules:**

* Heater can only be turned off if the temperature is above the minimum temperature.

**Use Case Specification**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Request temperature check. | System returns the temperature reading. Temperature is above the minimum. System turns the heater off. |
| 2 | Request temperature check, after 5mins. | System compares temperature reading to previous reading to see if the heater was turned off, thus affecting water temperature. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Temperature is above the minimum. | End Use Case. |
| A2 | System is unable to get the temperature reading. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |
| A3 | System is unable to turn off the heater. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Heater turned off. | The system turns the heater off. |
| Alt: No temperature reading. | The system is unable to get a temperature reading. |
| Alt: Heater does not turn off. | The system attempts to turn the heater off but after 5mins the temperature keeps going up. |
| Alt: Current temperature (of the water in the tank) is below the minimum. | The temperature is below the minimum. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Heater turned off | The temperature is above the minimum, and the temperature has stopped increasing. | The heater is off and the temperature of the water has stopped increasing. System continues monitoring the temperature. |
|  | Alt: No temperature reading. | The system does not get a temperature reading from the sensor. | Error message: “Unable to obtain temperature reading”; the system attempt to turn the heater off. Notification created and synced to phone (when connected). |
|  | Alt: Heater does not turn off. | System attempts to turn off the heater and checks the temperature after 5mins for a temperature increase. | Error message: “Heater cannot turn off”; notification created and synced to phone (when connected). |
|  | Alt: Current temperature (of the water in the tank) is below the minimum. | System compares the current temperature to the minimum. | System turns on the heater and continues monitoring the temperature checking if it goes above the minimum. |

#### Use Case Name: System Timer Turns Fan On

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Temperature range is set.
* Automatic temperature regulation is enabled.
* Temperature sensor is connected to the tank system.
* Tank is filled with water.

**Use Case Successful post condition(s)**:

* Fan turns on and starts cooling the water in the tank.

**Applicable Business Rules:**

* Fan can only be turned on if the temperature is above the maximum temperature.

**Use Case Specification**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Request temperature check. | System returns the temperature reading. Temperature is above the max range. System turns the fan on. |
| 2 | Request temperature check, after 5min. | System compares temperature reading to previous reading to see if the fan was turned on, thus affecting water temperature. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Temperature is above the max range. | End use case |
| A2 | System is unable to get temperature reading. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |
| A3 | System is unable to turn on the fan. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Fan turned on. | The system turns the fan on and starts cooling the water in the tank. |
| Alt: No temperature reading. | The system is unable to get a temperature reading from the sensor. |
| Alt: Fan does not turn on. | The system attempts to turn the fan on but after 5mins the temperature does not change. |
| Alt: Current temperature is above the maximum. | The temperature is above the maximum. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Fan turned on. | The temperature is above the maximum, and the temperature starts to decrease. | The heater is on and the temperature of the water is decreasing. System continues monitoring the temperature. |
|  | Alt: No temperature reading. | The system does not get a temperature reading from the sensor. | Error message: “Unable to obtain temperature reading”; the system does not attempt to turn the fan on. Notification created and synced to phone (when connected). |
|  | Alt: Fan does not turn on. | System attempts to turn on the fan and checks the temperature after 5mins for a temperature decrease. | Error message: “Fan cannot turn on”; notification created and synced to phone (when connected). |
|  | Alt: Current temperature is below the maximum. | System compares the current temperature to the maximum. | System turns off the fan and continues monitoring the temperature checking if it goes above the maximum. |

#### Use Case Name: System Timer Turns Fan Off

**Actor(s)**: System Timer

**Use Case Precondition(s)**:

* Temperature range is set.
* Automatic temperature regulation is enabled.
* Temperature sensor is connected to the tank system.
* Tank is filled with water.

**Use Case Successful post condition(s)**:

* Fan turns off and stops cooling the water in the tank.

**Applicable Business Rules:**

Fan can only be turned off if the temperature is below the minimum temperature.

**Use Case Specification**

**Main Flow:**

|  |  |  |
| --- | --- | --- |
|  | Actor(s): System Timer | System |
| 1 | Request temperature check. | System returns the temperature reading. Temperature is below the maximum. System turns the fan off. |
| 2 | Request temperature check, after 5 mins. | System compares temperature reading to previous reading to see if the fan was turned off, thus affecting water temperature. |

**Alternate Flows:**

|  |  |  |
| --- | --- | --- |
|  | Alternate Flow | Description |
| A1 | Temperature is below the maximum. | End Use Case. |
| A2 | System is unable to get the temperature reading. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |
| A3 | System is unable to turn off the heater. | System alerts user on their smartphone device through notification system. System sends a notification to the user. |

**Test Cases**

A. Test case Summary

|  |  |
| --- | --- |
| Scenario Name | Scenario Description |
| Main: Fan turned off. | The system turns the fan off. |
| Alt: No temperature reading. | The system is unable to get a temperature reading. |
| Alt: Fan does not turn off. | The system attempts to turn the fan off but after 5mins the temperature keeps going down. |
| Alt: Current temperature is below the maximum. | The temperature is below the maximum. |

B. Test Case Details

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Scenario | Data values / conditions being tested | Expected Result |
|  | Main: Fan turned off | The temperature is below the maximum, and the temperature has stopped decreasing. | The fan is off and the temperature of the water has stopped decreasing. System continues monitoring the temperature. |
|  | Alt: No temperature reading. | The system does not get a temperature reading from the sensor. | Error message: “Unable to obtain temperature reading”; the system attempt to turn the fan off. Notification created and synced to phone (when connected). |
|  | Alt: Fan does not turn off. | System attempts to turn off the fan and checks the temperature after 5mins for a temperature increase. | Error message: “Fan cannot turn off”; notification created and synced to phone (when connected). |
|  | Alt: Current temperature is above the maximum. | System compares the current temperature to the maximum. | System turns the fan on and continues monitoring the temperature checking if it goes below the maximum. |