**1. Introduction**

**1.1 Purpose of Document**

There is a need to develop an application to assist in swimming pool club management using

modern technologies such as Python and MySQL. Too many pool clubs around the country have been plagued with mismanagement, causing irregular hours for pool staff and the management of chemicals (like the pH levels) to be ignored. Many pool clubs exceed their budget due to overusing pool chemicals, running short of pool chemicals, and mismanaging lifeguard scheduling. Additionally, pools are frequently inspected by local health departments. Many times, pools are not prepared for these inspections due to incomplete or missing documentation concerning water testing and licensing certifications resulting in temporary closure. The goal of this project is to create a database using modern technologies to keep track of complex pool tasks, water quality, maintenance, documentation, and scheduling so that financial and staff availability concerns will become a thing of the past.

**1.2 Project Summary**

|  |  |
| --- | --- |
| **Project Name:** | Pool Club Management Application |
| **Project Manager:** | Evan D. Lyle |
| **Project Advisor:** | Prof. Jal Irani |
| **Responsible Users:** | Community Pool Clubs, Hotels |

**1.3 Background**

Too many pool clubs around the country have been plagued with mismanagement, causing irregular hours for pool staff and the management of chemicals (like the pH levels) to be ignored. Many pool clubs exceed their budget due to overusing pool chemicals, running short of pool chemicals, and mismanaging lifeguard scheduling. Additionally, pools are frequently inspected by local health departments. Many times, pools are not prepared for these inspections due to incomplete or missing documentation concerning water testing and licensing certifications resulting in temporary closure. The goal of this project is to create a database using modern technologies to keep track of complex pool tasks, water quality, maintenance, documentation, and scheduling so that financial and staff availability concerns will become a thing of the past.

**1.4 Project Scope**

The scope of this project is an application that helps to keep track of your

resources. For example, monitoring and tracking levels of chlorine and pH on a daily basis and total hardness, total alkalinity, and cyanuric acid levels on a weekly basis. For pool maintenance these include the amount of chemicals used versus the amount of chemicals that are in-stock; the ability to track and input the amount and type of chemicals recently purchased/stocked; and reminders of which chemicals need to be restocked based on scheduled use and if chemicals fall below a certain threshold. Another function of the database is the ability to add, edit, delete, and schedule for lifeguards and employees to oversee and manage the pool.

**1.5 Software Purpose**

**1.5.1 Users**

The target users of this pool application project are those who manage and work with different aspects of community pools.

Pool Managers:

The ones in charge of the pool club. They oversee all of the pool operations and can freely add and remove various types of users.

Pool Operator/Lifeguard:

The pool operator is the second in command. They are in charge of helping the Pool Manager(s) with the various tasks, including scheduling. In addition, pool operators double as lifeguards.

Lifeguard:

The lifeguards add and regulate the various chemicals to keep the levels of chlorine, pH, hardness, alkalinity, and cyanuric acid at the appropriate levels.

Employees (general purpose):

These users consist of other types of employees, who are limited to things like generating and viewing schedules.

**1.5.2 Location**

The software will be available to any potential customer using the local MySQL database, available for download on the Internet to various operating systems (Windows, Mac, Linux).

**1.5.3 Responsibilities**

The primary responsibilities of the application are as follows:

* Sort and edit schedules for the various staff members.
* Manage the water quality by measuring the readings.
* Manage the different pool chemicals.
* Generate reports of chemical readings in a document format.
* Generate schedules in a document format.
* A secure login system, with as little injection bypassing as possible.

Other desired features of the application include:

* An easy-to-use and intuitive layout.
* A collapsible readings system that displays current readings without having to change screens.
* Account for different permissions depending on the type of staff account.

**1.5.4 Need**

Most, if not all, pool clubs across the country will benefit from this pool club management system. There’s a fair bit of mismanagement in most of these clubs, so this would be a godsend to those struggling with juggling the different resources (be they chemicals, staff members, or even time management on the various tasks).

**1.6 Overview of Document**

The rest of this document gives the detailed specifications for the new sales system. It is organized as follows:

* Section 2: Functional Objectives  
  Each objective gives a desired behavior for the system, a business justification, and a measure to determine if the final system has successfully met the objective. These objectives are organized by priority. In order for the new system to be considered successful, all high priority objectives must be met.
* Section 3: Non-Functional Objectives  
  This section is organized by category. Each objective specifies a technical requirement or constraint on the overall characteristics of the system. Each objective is measurable.
* Section 4: Context Model  
  This section gives a text description of the goal of the system, and a pictorial description of the scope of the system in a context diagram. Those entities outside the system that interact with the system are described.
* Section 5: Use Case Model  
  The specific behavioral requirements of the system are detailed in a series of use cases. Each use case accomplishes a business task and shows the interaction between the system and some outside actor. Each use case is described with both text and an interaction diagram. An interface prototype is also shown. The system use case diagram depicts the interactions between all use cases and system actors.

**2. Functional Objectives**

**2.1 High Priority**

1. This system will be able to allow access to record data of the appropriate schedules for the user clearance.
2. The system will provide access to both daily and weekly readings.
3. The system shall follow the CRUD design philosophy in terms of looks and functionality.
   1. C = Create: Users will be able to create records of readings, tasks, and staff members as appropriate for their staff clearance.
   2. R = Read: Users will easily be to display data about readings, tasks, and staff members. This will be possible in both table format and generated reports.
   3. U = Update: Users will be able to update records of readings, tasks, and staff members as appropriate for their staff clearance.
   4. D = Delete: Users will be able to remove records of readings, tasks, and staff members as appropriate for their staff clearance.
4. The system will have a secure login and authentication process that prevents tampering.

**2.2 Medium Priority**

1. Summaries of daily and weekly readings will be accessible via a collapsible display without having to change the user’s location in the application.
2. The system will have a lockdown security feature should the user attempting to access the application and database fail up to five times.
3. Tutorials and references for how the system works will be available to users via the Help drop down menu.

**2.3 Low Priority**

1. Tutorials and references for how the system works will be available to users via the ‘Help’ drop down menu.
2. There will be a draft option for records just in case the user needs to double check the accuracy of the data being inputted before finalizing a record.

**3. Non-Functional Objectives**

**3.1 Reliability**

* The system shall be completely operational at least 98% of the time.
* Down time after a failure will be restored by backup.

**3.2 Usability**

* A user should be able to use the software in their job after 3 days of training.
* A user who already knows what they are interested in should be able to locate and view that screen in 3 to 6 seconds.
* The number of screens navigated to access information from the top of the window should not exceed 3 to 4 clicks.

**3.3 Performance**

* The software should be able to support one login at a time.
* The mean time to view a screen shall not exceed 3 seconds.
* The mean time to download and view a document in PDF format shall not exceed 10 seconds.

**3.4 Security**

* The system shall provide password protected access to the software.

**3.5 Supportability**

* The software should be able to accommodate new updates without major reengineering.

**3.6 User Documentation and Help**

* The software shall provide a tab that explains how to navigate the program. This feature should be customized based on what screens that user is allowed to access.
* This help screen should be accessible from all other screens.

**3.7 Interfaces**

The software must interface with the local MySQL database.

**4. The Context Model**

**4.1 Goal Statement**

The goal of the software is to provide community pool clubs with a database management software that helps organize complex pool tasks, water quality, maintenance, documentation, and scheduling to reduce financial and staff availability concerns.

**4.2 Context Diagram**



**4.3 System Externals**

Pool Manager

The one in charge of managing the pool supplies and the various staff employed at the pool club. They have overall responsibility for the various records and activities that go on and are thus considered the product owner.

Pool Operator/Lifeguard

The assistant of the pool manager. They set most of the schedules and oversee the supplies, as well as acting as a lifeguard.

Lifeguard

The lifeguards are mostly in charge of the records regarding water quality and the pool chemical supply. They can also generate reports.

Employee

This category of staff refers to general-purpose employees. The most they can do is generate reports and view the schedule for work shifts.

**5. The Use Case Model**

**5.1 Software Use Case Diagrams**

A screen shot of a phone

Description automatically generatedA screen shot of a phone

Description automatically generatedA screen shot of a phone

Description automatically generatedA black background with a black square

Description automatically generated

**5.2 Use Case Descriptions (for selected cases)**

Notes:

* For all use cases, the user can cancel the use case at any step that requires user input. This action ends the use case. Any data collected during that use case is lost.
* All use cases involve the user being logged in. Therefore, all login sessions are updated during the use case to reflect the navigation paths through the use case.

**Login User**

|  |  |
| --- | --- |
| Use Case Name: | Login User |
| Summary: | In order to get personalized or restricted information, place orders or do other specialized transactions a user must login so that that the systemsoftware can determine their access level. |
| Basic Flow: | 1. The use case starts when a user indicates that they want to login. 2. The software requests the username and password. 3. The user enters their username and password. 4. The software verifies the username and password against all registered users. 5. The software starts a login session and goes to the home screen, which may vary depending on the user's type of account. |
| Alternative Flows: | Step 4:  Should the username be invalid, return to step 2.  Step 4:  Should the password be invalid, the software requests that the user re-enter the password. When the user enters another password the use case continues with step 4 using the original username and new password. |
| Extension Points: | none |
| Preconditions: | The user is registered. |
| Postconditions: | The user can now obtain data and perform functions according to their registered access level. |
| Business Rules: | Some data and functions are restricted to certain types of users or users with a particular access level. |

**Add Staff to Database**

|  |  |
| --- | --- |
| Use Case Name: | Add Staff to Database |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to add a staff member to the database, which will also make them a user of the software. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard user wants to add a new staff member to the roster. 2. The user clicks on the Staff Organization option from the home screen menu. 3. The software displays all staff members registered in the database. 4. The user inputs the appropriate data into the different fields (ID num, First Name, Last Name, Hire Date, Password, and User Type; leave Separation Date and Reason for Separation blank). 5. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | *Edit Staff in Database; Delete Staff from Database* |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the new staff member (who is also a user) has been added to the database. |

**Edit Staff in Database**

|  |  |
| --- | --- |
| Use Case Name: | Edit Staff in Database |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to change the details of a staff member in the database. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard needs to change the details of an existing staff member. 2. The user clicks on the Staff Organization option from the home screen menu. 3. The software displays all staff members registered in the database. 4. The user selects a staff member in the database to change the details of. The entry fields will automatically display the data currently registered in the database. 5. The user makes a change in one of the existing fields (ID num, First Name, Last Name, Hire Date, Password, and User Type; leave Separation Date and Reason for Separation blank). 6. The user clicks the “Update Record” button. |
| Alternative Flows: | 1. If the staff member is leaving the pool club, fill in both the Separation Date and the Reason for Separation fields. |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the record for the staff member has been successfully modified. |

**Delete Staff from Database**

|  |  |
| --- | --- |
| Use Case Name: | Delete Staff from Database |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to remove a staff member from the database, thereby revoking their user privileges. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard user wishes to delete a staff member from the database. 2. The user clicks on the Staff Organization option from the home screen menu. 3. The software displays all staff members registered in the database. 4. The user selects the staff member to be removed from the database. The entry fields will automatically display the data currently registered in the database. 5. The user clicks the “Remove Record” button. 6. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 7. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. The staff member to be deleted has been either absent for 10+ years or in the case of allocated database storage being 80% full, 3–5+ years. |
| Postconditions: | The application will display a pop-up message informing the user that the staff member(s) has/have been removed from the database. |

**Schedule Maintenance**

|  |  |
| --- | --- |
| Use Case Name: | Schedule Maintenance |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to set a maintenance schedule by sorting maintenance tasks. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard needs to sort maintenance tasks. 2. The user clicks on the Schedule Maintenance option from the home screen menu. 3. The software displays all maintenance tasks registered in the database. |
| Alternative Flows: | none |
| Extension Points: | *Add Maintenance Task; Edit Maintenance Task; Remove Maintenance Task* |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | Tasks are displayed and can be added to, edited, or removed. |

**Add Maintenance Task**

|  |  |
| --- | --- |
| Use Case Name: | Add Maintenance Task |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to add a maintenance task into the database. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard needs to add a new maintenance task. 2. The user inputs the appropriate data into the different fields (ItemNumber, Date, Task, Staff Rep., Status). 3. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is on the Schedule Maintenance screen. |
| Postconditions: | The application will display a pop-up message informing the user that the maintenance task has been added to the database. |

**Edit Maintenance Task**

|  |  |
| --- | --- |
| Use Case Name: | Edit Maintenance Task |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to edit an existing maintenance task. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard needs to modify an existing maintenance task. 2. The user clicks on one of the tasks they wish to change the details of. The entry fields will automatically display the data currently registered in the database. 3. The user makes a change in one of the existing fields (ItemNumber, Date, Task, Staff Rep., Status). 4. The user clicks the “Update Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is on the Schedule Maintenance screen. |
| Postconditions: | The application will display a pop-up message informing the user that the record for the maintenance task has been successfully modified. |

**Remove Maintenance Task**

|  |  |
| --- | --- |
| Use Case Name: | Remove Maintenance Task |
| Summary: | This use case allows the Pool Manager and a Pool Operator/Lifeguard to remove a maintenance task from the database. |
| Basic Flow: | 1. The use case starts when the Pool Manager or a Pool Operator/Lifeguard needs to remove a maintenance task. 2. The user selects the maintenance task to be removed from the database. The entry fields will automatically display the data currently registered in the database. 3. The user clicks the “Remove Record” button. 4. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 5. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is on the Schedule Maintenance screen and operations are considered over for the year or the database is at least 80% full. |
| Postconditions: | The application will display a pop-up message informing the user that the maintenance task(s) has/have been removed from the database. |

**Add Daily Reading**

|  |  |
| --- | --- |
| Use Case Name: | Add Daily Reading |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to add their daily readings for the pool’s water quality to the database. |
| Basic Flow: | 1. The use case starts when the user needs to add their daily readings from the pool to the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Daily Readings from the options menu. 4. The software displays all of the current daily readings registered in the database. 5. The user inputs the appropriate data into the different fields (Day\_ID, Date, Time, pH, Total Chlorine, Free Chlorine, Combined Chlorine). 6. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the daily reading of the pool’s water quality has been added to the database. |

**Edit Daily Reading**

|  |  |
| --- | --- |
| Use Case Name: | Edit Daily Reading |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to edit an existing daily reading of the pool’s water quality. |
| Basic Flow: | 1. The use case starts when the user needs to correct an existing daily reading in the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Daily Readings from the options menu. 4. The software displays all of the current daily readings registered in the database. 5. The user makes a change in one of the existing fields (Day\_ID, Date, Time, pH, Total Chlorine, Free Chlorine, Combined Chlorine). 6. The user clicks the “Update Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the daily reading record of the pool’s water quality has been successfully modified. |

**Remove from Daily Readings**

|  |  |
| --- | --- |
| Use Case Name: | Remove from Daily Readings |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to remove an existing daily reading of the pool’s water quality from the database. |
| Basic Flow: | 1. The use case starts when the user needs to remove a daily reading from the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Daily Readings from the options menu. 4. The software displays all of the current daily readings registered in the database. 5. The user selects the daily reading to be removed from the database. The entry fields will automatically display the data currently registered in the database. 6. The user clicks the “Remove Record” button. 7. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 8. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the daily reading(s) of the pool’s water quality has/have been removed from the database. |

**Add Weekly Reading**

|  |  |
| --- | --- |
| Use Case Name: | Add Weekly Reading |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to add their weekly readings for the pool’s water quality to the database. |
| Basic Flow: | 1. The use case starts when the user needs to add their weekly readings from the pool to the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Weekly Readings from the options menu. 4. The software displays all of the current weekly readings registered in the database. 5. The user inputs the appropriate data into the different fields (Week\_ID, Date, Time, pH, Total Hardness, Total Alkalinity, CYA (stabilizer)). 6. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the weekly reading of the pool’s water quality has been added to the database. |

**Edit Weekly Reading**

|  |  |
| --- | --- |
| Use Case Name: | Edit Weekly Reading |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to edit an existing weekly reading of the pool’s water quality. |
| Basic Flow: | 1. The use case starts when the user needs to correct an existing weekly reading in the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Weekly Readings from the options menu. 4. The software displays all of the current weekly readings registered in the database. 5. The user makes a change in one of the existing fields (Week\_ID, Date, Time, pH, Total Hardness, Total Alkalinity, CYA (stabilizer)). 6. The user clicks the “Update Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the weekly reading record of the pool’s water quality has been successfully modified. |

**Remove from Weekly Readings**

|  |  |
| --- | --- |
| Use Case Name: | Remove from Weekly Readings |
| Summary: | This use case allows a user (Pool Manager, Pool Operator/Lifeguard, or Lifeguard) to remove an existing weekly reading of the pool’s water quality from the database. |
| Basic Flow: | 1. The use case starts when the user needs to remove a weekly reading from the database. 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Weekly Readings from the options menu. 4. The software displays all of the current weekly readings registered in the database. 5. The user selects the weekly reading to be removed from the database. The entry fields will automatically display the data currently registered in the database. 6. The user clicks the “Remove Record” button. 7. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 8. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager, Pool Operator/Lifeguard, or Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the weekly reading(s) of the pool’s water quality has/have been removed from the database. |

**Generate Daily/Weekly Logs Report**

|  |  |
| --- | --- |
| Use Case Name: | Generate Daily/Weekly Logs Report |
| Summary: | This use case allows a user to compile the daily readings and weekly readings of the pool’s water quality into a single report. |
| Basic Flow: | 1. The use case starts when the user needs to generate a report of the water quality readings (both daily and weekly). 2. The user clicks on the Water Quality Readings option from the home screen menu. 3. The user clicks on Daily/Weekly Logs from the options menu. 4. Fill in the date relevant to the log you wish to generate, then press the “Generate Log” button. 5. The software generates a view of the log. 6. The user clicks the “Download” button to save the log as a .pdf file. 7. [Optional] The user clicks the “Print” button to have a physical copy of the generated report. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in. |
| Postconditions: | The Water Quality Report containing both daily and weekly logs is generated and on-hand. |

**Add Chemical to Pool Chemical Supply**

|  |  |
| --- | --- |
| Use Case Name: | Add Chemical to Pool Chemical Supply |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to add a supply of a chemical to the database. |
| Basic Flow: | 1. The use case starts when the user needs to add a chemical record to the database. 2. The user clicks on the Pool Chemical Supply option from the home screen menu. 3. The software displays all of the current pool chemicals registered in the database. 4. The user inputs the appropriate data into the different fields (Chemical Name, Amount Purchased, Date Purchased, Amount Used, Date Used, Amount On-hand). 5. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the supply of a specific pool chemical has been added to the database. |

**Edit Chemical from Pool Chemical Supply**

|  |  |
| --- | --- |
| Use Case Name: | Edit Chemical from Pool Chemical Supply |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to edit the record of an existing chemical in the database. |
| Basic Flow: | 1. The use case starts when the user needs to update the supply of a chemical recorded in the database. 2. The user clicks on the Pool Chemical Supply option from the home screen menu. 3. The software displays all of the current pool chemicals registered in the database. 4. The user makes a change in one of the existing fields (Chemical Name, Amount Purchased, Date Purchased, Amount Used, Date Used, Amount On-hand). 5. The user clicks the “Update Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the record of a pool chemical’s supply has been successfully modified. |

**Remove Chemical from the Pool Chemical Supply**

|  |  |
| --- | --- |
| Use Case Name: | Remove Chemical from the Pool Chemical Supply |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to remove an existing supply of pool chemicals from the database. |
| Basic Flow: | 1. The use case starts when the user needs to remove a record of a supply for one of the pool chemicals from the database. 2. The user clicks on the Pool Chemical Supply option from the home screen menu. 3. The software displays all of the current pool chemicals registered in the database. 4. The user selects the chemical supply to be removed from the database. The entry fields will automatically display the data currently registered in the database. 5. The user clicks the “Remove Record” button. 6. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 7. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the supply of (a) pool chemical(s) has/have been removed from the database. |

**Generate Water Quality Regulation Report**

|  |  |
| --- | --- |
| Use Case Name: | Generate Water Quality Regulation Report |
| Summary: | This use case allows a user to compile the water quality readings and the pool chemical supply into a water quality regulation report. |
| Basic Flow: | 1. The use case starts when the user needs to generate a report for water quality regulation. 2. The user clicks on the Water Quality Regulation Report option from the home screen menu. 3. Fill in the date relevant to the log you wish to generate, then press the “Generate Log” button. 4. The software generates a view of the log. 5. The user clicks the “Download” button to save the log as a .pdf file. 6. [Optional] The user clicks the “Print” button to have a physical copy of the generated report. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in. |
| Postconditions: | The Water Quality Regulation Report has been compiled, which will inform the user of what chemicals and readings need to be checked on in order to reach and/or maintain acceptable water quality levels. |

**Add Staff to Staff Scheduling**

|  |  |
| --- | --- |
| Use Case Name: | Add Staff to Staff Scheduling |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to assign a task to a staff member. |
| Basic Flow: | 1. The use case starts when the user needs to add to the schedule for their staff. 2. The user clicks on the Staff Scheduling option from the home screen menu. 3. The user clicks the Add/Edit/Remove option from the options menu. 4. The software displays all of the scheduled work shifts registered in the database. 5. The user inputs the appropriate data into the different fields (Date, Shift, Manager/Pool Operator, Lifeguards, Admin). 6. The user clicks the “Add Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that a staff member has been assigned a work shift in the database. |

**Edit Staff in Staff Scheduling**

|  |  |
| --- | --- |
| Use Case Name: | Edit Staff in Staff Scheduling |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to edit the record of an existing work shift in the database. |
| Basic Flow: | 1. The use case starts when the user needs to modify the schedule for their staff. 2. The user clicks on the Staff Scheduling option from the home screen menu. 3. The user clicks the Add/Edit/Remove option from the options menu. 4. The software displays all of the scheduled work shifts registered in the database. 5. The user selects a work shift and modifies the appropriate data fields (Date, Shift, Manager/Pool Operator, Lifeguards, Admin). 6. The user clicks the “Update Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that a staff member’s work shift in the schedule has been updated in the database. |

**Remove Staff from Staff Scheduling**

|  |  |
| --- | --- |
| Use Case Name: | Remove Staff from Staff Scheduling |
| Summary: | This use case allows a user (Pool Manager or Pool Operator/Lifeguard) to remove an existing staff member’s work shift from the schedule in the database. |
| Basic Flow: | 1. The use case starts when the user needs to remove a scheduled task from the database. 2. The user clicks on the Staff Scheduling option from the home screen menu. 3. The user clicks the Add/Edit/Remove option from the options menu. 4. The software displays all of the scheduled tasks registered in the database. 5. The user selects a scheduled task to be removed from the database. The entry fields will automatically display the data currently registered in the database. 6. The user clicks the “Remove Record” button. 7. There is a pop-up message asking if the user is sure they wish to delete the record, with a “Yes” and “No” button. 8. The user clicks “Yes” to proceed with the record’s deletion. |
| Alternative Flows: | Removing Multiple Records:   1. The user selects multiple records. Only the most recently selected record will be displayed in the entry fields. 2. The user clicks the “Remove Multiple Records” button. 3. There is a pop-up message asking if the user is sure they wish to delete the selected records, with a “Yes” and “No” button. 4. The user clicks “Yes” to proceed with the deletion of the selected records.   Cancelling the Deletion:   1. The user clicks the “No” button, cancelling the operation that would delete the selected record(s). |
| Extension Points: | none |
| Preconditions: | The user is logged in and has a User Type of Pool Manager or Pool Operator/Lifeguard. |
| Postconditions: | The application will display a pop-up message informing the user that the specified work shift(s) for (a) staff member(s) has/have been removed from the schedule in the database. |

**View Work Shifts from Staff Scheduling**

|  |  |
| --- | --- |
| Use Case Name: | View Work Shifts from Staff Scheduling |
| Summary: | This use case allows for the user to simply view the schedule. |
| Basic Flow: | 1. The use case starts when a user wishes to view the schedule without making any changes to it. 2. The user clicks on the Staff Scheduling option from the home screen menu. 3. The user clicks the View Schedules option from the options menu. 4. The software displays all of the scheduled work shifts registered in the database. 5. [Optional] The user can search for specific shifts by filling in the text entry fields and then clicking on the “Search Record” button. |
| Alternative Flows: | none |
| Extension Points: | none |
| Preconditions: | The user is logged in. |
| Postconditions: | The user has access to the work shifts that need to be fulfilled. |