**Investigation**

**Project Outline**

I am going to design a new **Lesson Management system** for a Swimming Club to help manage data stored for their swimmers. This system is going to be designed in Python using a Tkinter Graphical User Interface (GUI) and a SQL oriented database. The SQL database will store Swimmer names, Levels, Parent information etc. I will also be using an Object Orientated Approach (OOP) for easier implementations of functionalities within the scope of the project.

**Current Solution**

The current system for Lisburn LeisurePlex involves:

* Logging in the system using a **4-digit code**, a second login screen asks for the **user’s memorable information** (which they decide)**.**
* They choose their **“Location and Activity” i.e.,** **LVPL – Swimming**
* They are brought to their **list of classes for the day** (all teachers can see other classes on that day)
* Each class has a **register and a syllabus**, the **Level** of the class, **Teacher** of the class, **Time** of class and **Number of Students.**
* At the **end of each month** the Manager must look through all the data, then print out the data to be **stored in the File room**. This is then manually sorted based on: **Date of created data, Level of Class, Year.** All **data** that is stored on the system is then **deleted**.

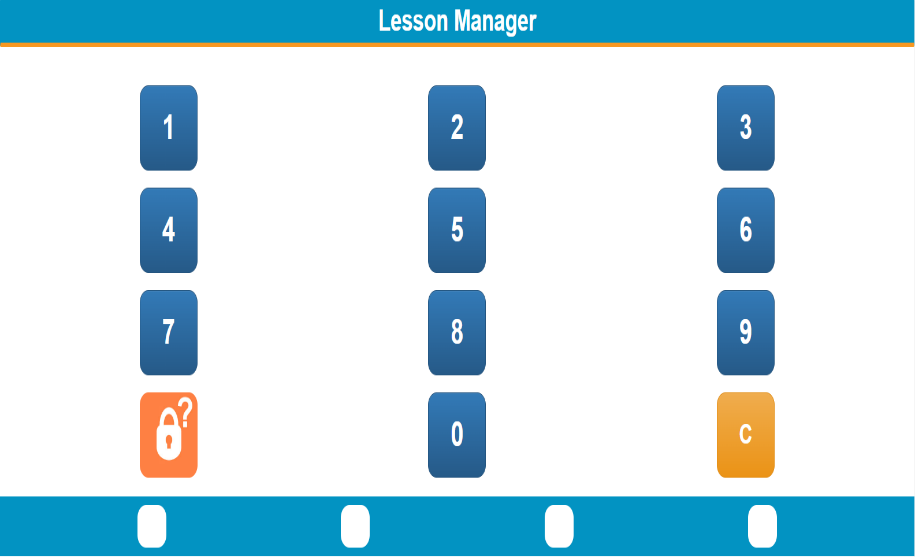
**Limitations of the current system**

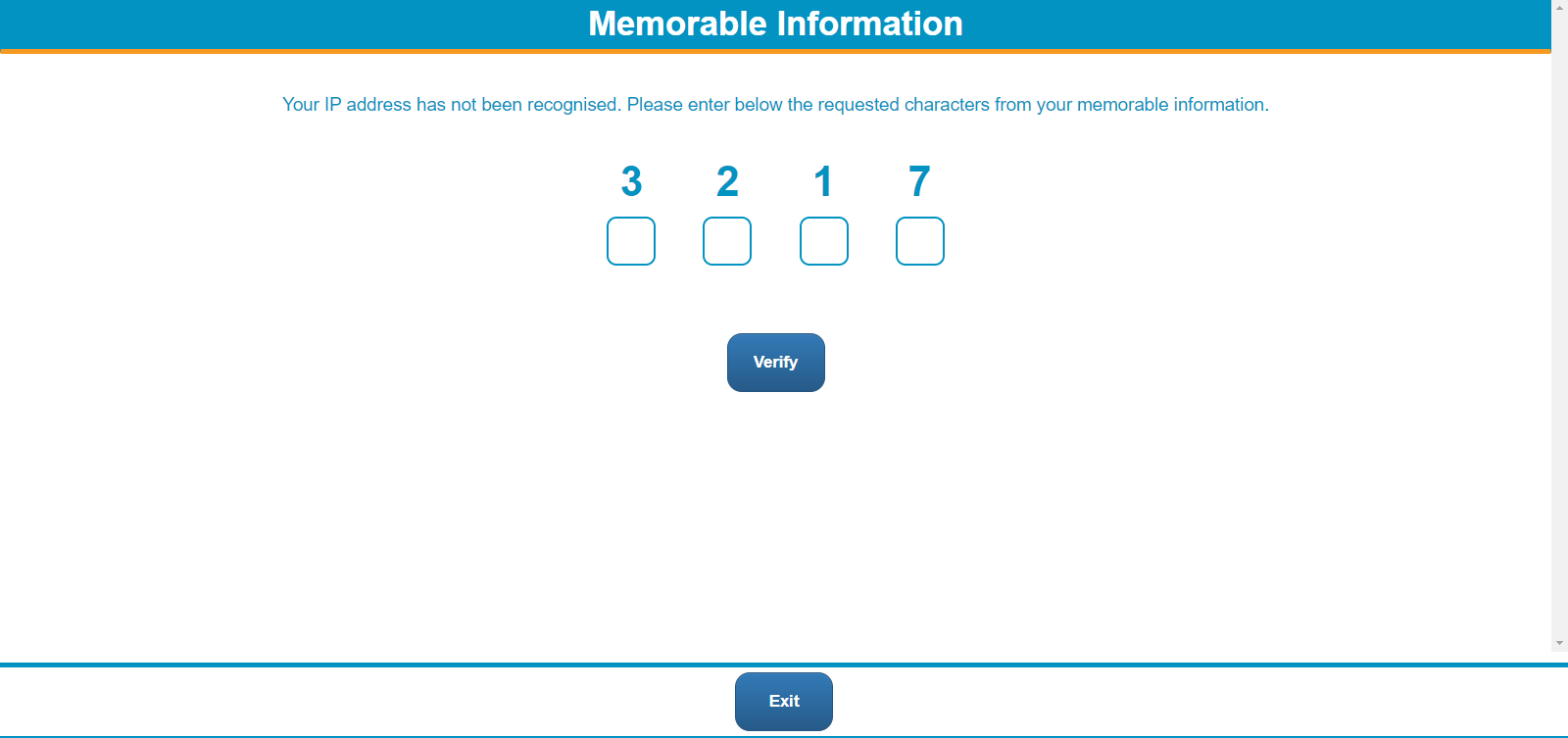
* Data is **difficult to backup** – All data is printed off and stored physically in files. (This means that these files can be damaged, lost or simply mishandled during this period).
* Data is **difficult to edit/remove** - Since all data is stored physically, any data to be changed or amended must be printed out again. This is tedious and a waste of resources.
* **Damaged files** – due to the humidity and moisture within the swimming club, the **files can be damaged**. This means that new data must be provided ASAP otherwise **data** on those swimmers **will be lost**. There is currently no way around this issue apart from constantly surveilling the File room.

**Desk based Research**

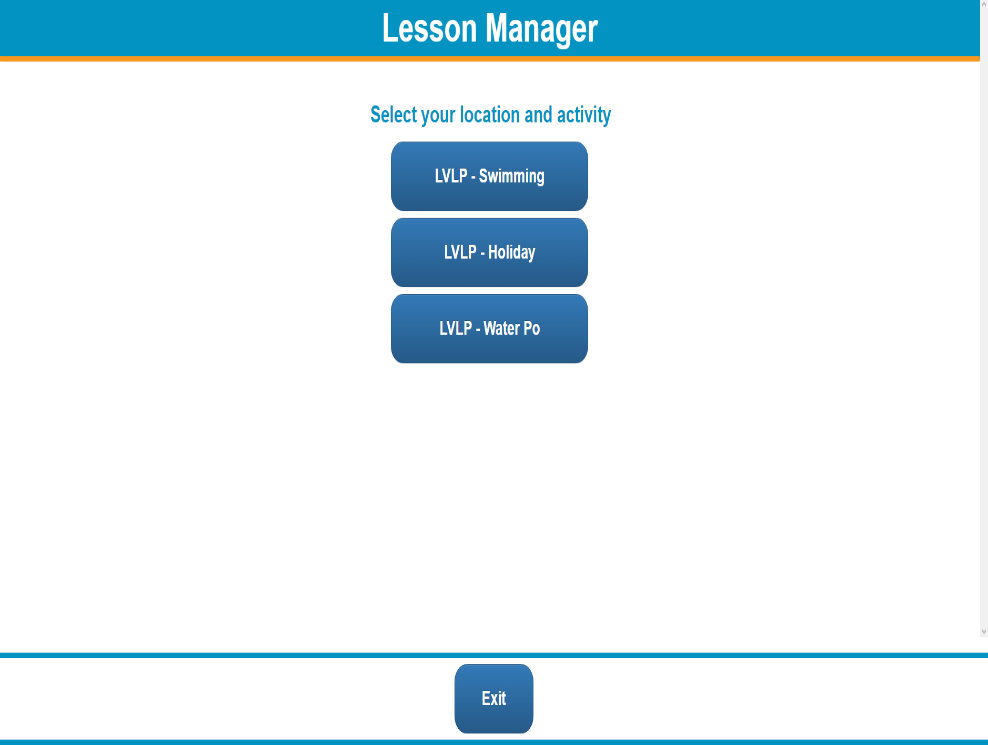
I have found 2 solutions which are similar in design to one another for the Lesson Manager system. I plan to keep this design approach and add additional functionalities that the Manager asked for.

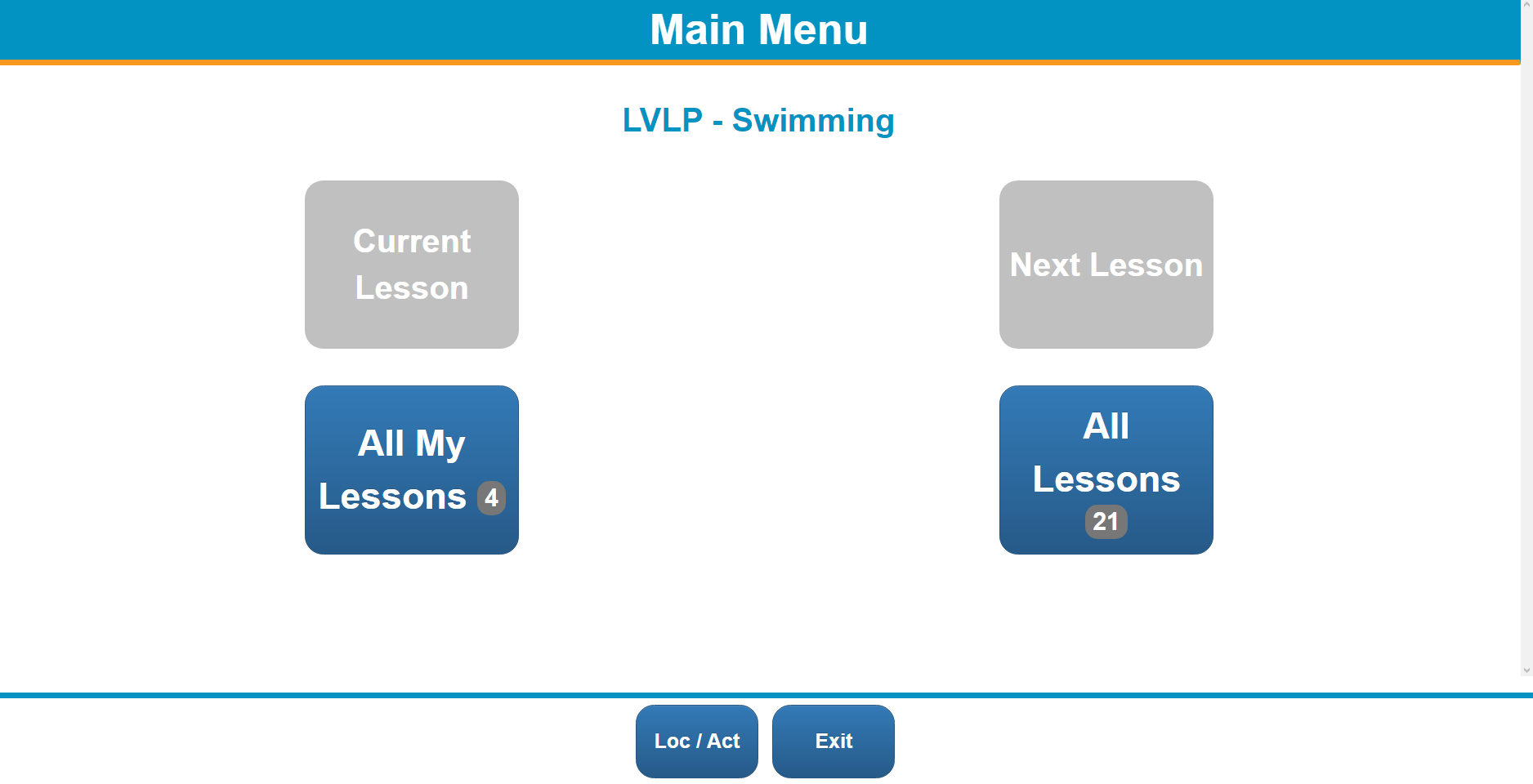
1. Morleisure

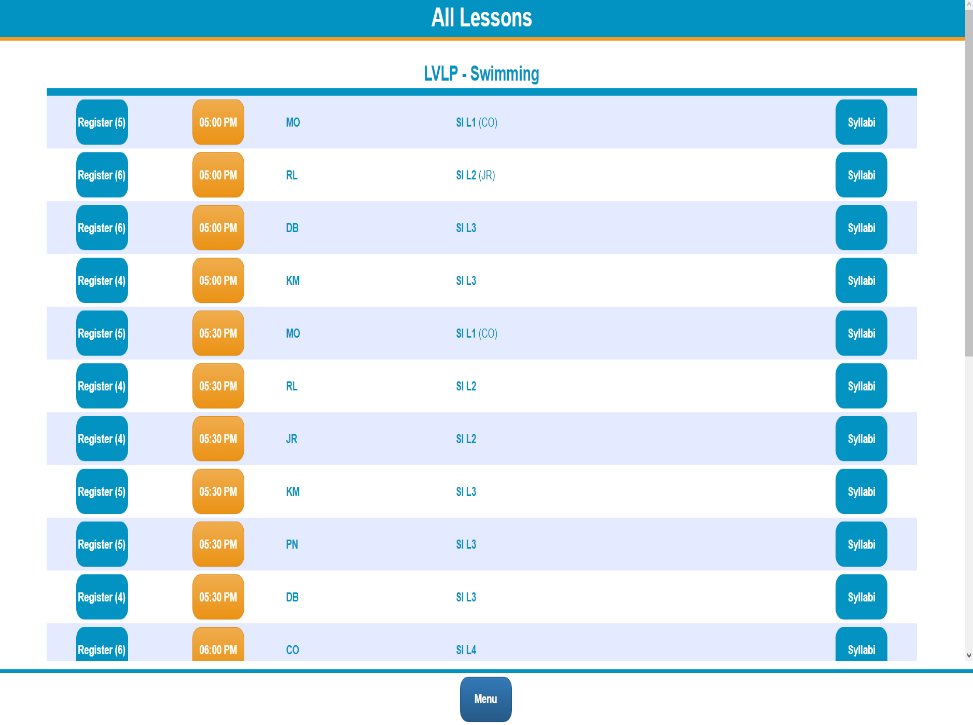
This is the **Login screen** for the Lesson Manager. Their Lesson Manager is web-based. I stated I will not be implementing this into the project. It is a **”C Button”** which **clears the input** for the password”**?” button** for **forgotten passwords**. Each password has **4 attempts** to login.

This is one of the **security measures** put in place by Morleisure.

Each Teacher is asked to construct their own **“Memorable Information"** that is **personalized** to them. In this example the Teacher is asked to enter the **third, second, first and seventh characters** of their **“Memorable Information.”**

This is the **”Selection Screen”** where Teachers choose the **field/time** in which **they teach** e.g. **A Swimming Teacher would choose LVLP – Swimming etc.**  This is currently not relevant to my current project as I would only be working on a Lesson Manager for **”LVLP – Swimming Teachers”**

This is the **”Lesson Selection”** screen. This has a few features that the current Manager would like to implement such as the **”Current Lesson” function**, which brings up the Registry for that set Teacher’s class at that time. All Lessons are available for all Teachers to view, and each box has several appearances: 1. **Grey** - button can’t be used 2. **Blue** – can be use. 3. **Blue with Num.** - shows the number of classes available for that specific function i.e. **”All Lessons”** button has **21 classes.**



This is the UI for the Lessons**.** This shows: 1. The **time** of the class. 2. The **no. of swimmers** in the class. 3. **Syllabus** for the class. 4. **Level** of class/Initials of **Teacher.**  I would like to implement this style of UI as it makes sense for what the Manager is looking for.

This is the UI for the Registry. The **Red name** signifies that the swimmer has **SEND** (Special Educational Needs).

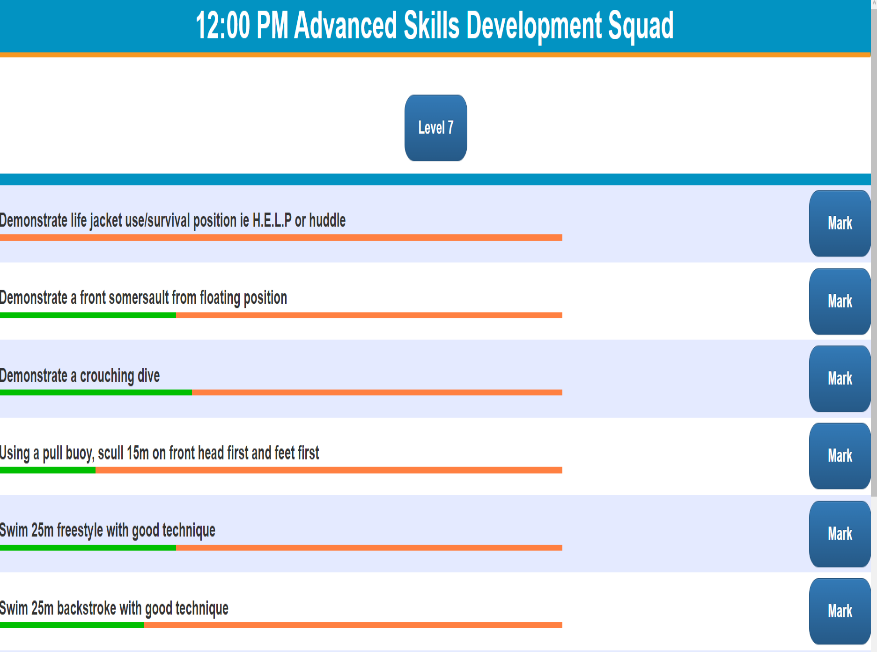
**Camera icon** shows if parents **consent** for photos being taken

**Coloured boxes** show if the swimmer was marked: **Present, Absent or Unmarked.**

This is the UI for **Moving Up** swimmers to each Level.

This is an abstract version of the **Moving up** process as normally **a review of the swimmer’s performance** is attached with the **Swimmer** and sent to the parents.

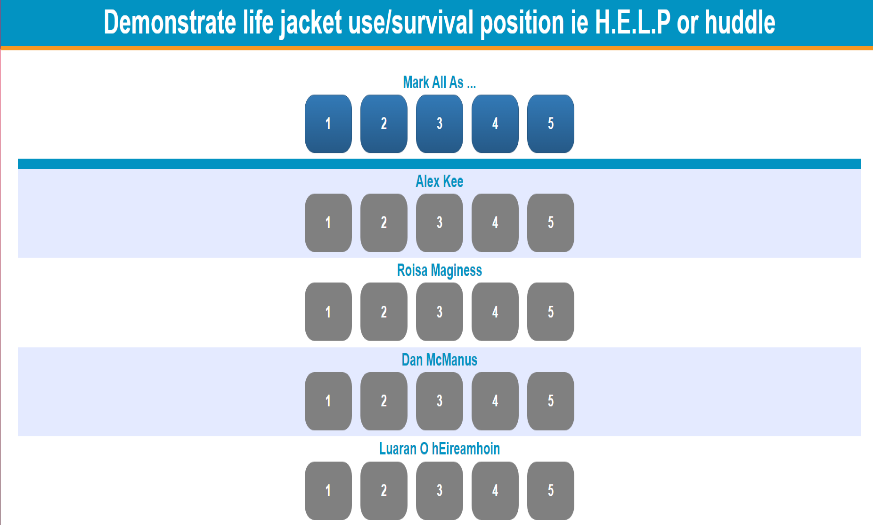
I might decide to **implement** this feature in its **abstract** form as it is **easier to implement/test.**

This is the “**Syllabus**” section.

This gives each **area** which the swimmer must practice to pass **the Level.**

The bar fills up from Red to Green depending on the marks each swimmer has achieved in each area.

If all swimmers have achieved full marks in all areas, then the bar is fully green.

This is within the “Syllabus” section where each swimmer is **marked out of 5.**

**4 and 5** are **passes** while the **1-3** are **fails.**

1. SwimSoft

A screenshot of a login page

Description automatically generatedThis is the **login screen** for SwimSoft’s Lesson Manager.

This solution is also **web-based** but instead uses a **Username and Password** approach for Security.

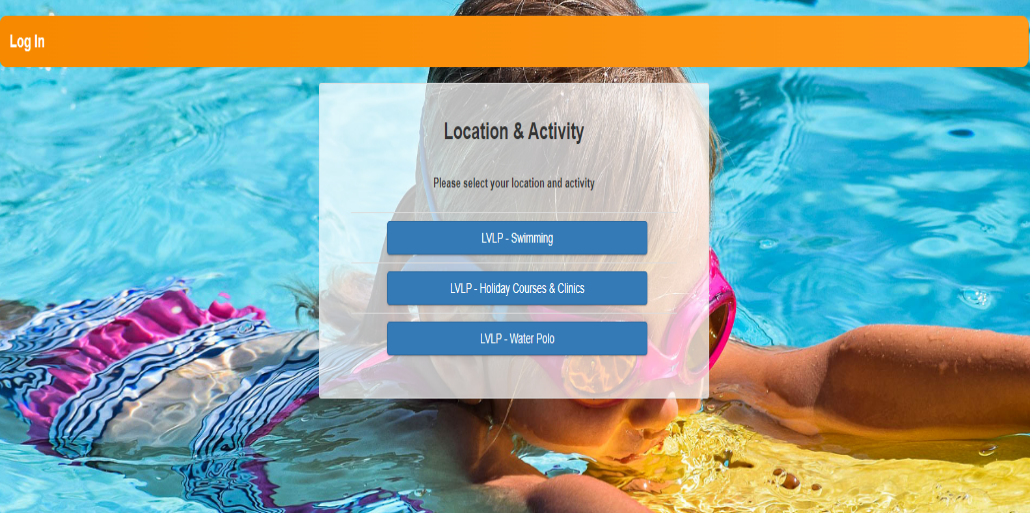
Each Teacher can select their **own** **Username and Password.**

A child swimming in a pool

Description automatically generated

This is another **security measure** that SwimSoft have used which is like Morleisure.

Each teacher creates their own **“Memorable Information”** to add an extra layer of security.

 This is the ”Selection **Screen”** where Teachers choose the **field/time** in which **they teach.**

Again, I will not be implementing this to my current project as my priority is to only get the ‘Swimming’ function working.

A screenshot of a computer

Description automatically generatedThis is the “**Selection Screen”** for the **‘Swimming Lessons’** there is no functions which show your ‘Current Lesson’ etc unlike Morleisure. This is because each Teacher has access only to their account.

This isn’t necessarily an issue, however, it **prevents other Teachers** to do things like **accessing** another Teacher’s **account** if they are **off sick**. So, I will create a ‘**Selection Screen’** like Morleisure to prevent this.

A screenshot of a computer

Description automatically generatedThis is the **‘Mark Attendance’** function which allows the Teacher to **filter** their Swimmer’s class based on: **Day, Class Level, Times and Course Number**. You can also ‘Clear’ this filter.

The ‘**Mark Attendance’** screen also links with the **‘Registry’**. I might **implement** this feature as it allows for **reusability** within the system.

The **Registry** also **changes colour** based on the **level of completion**.

A screen shot of a student attendance

Description automatically generatedThis is the **‘Registry’** that stores the attendance of all the swimmers. There is a ‘store’ function which saves the current attendance info selected.

The only **flaw** with this approach is that there is no way to mark **if a swimmer is absent or off sick.**

**A screen shot of a computer

Description automatically generated**This is the **‘Student Progress’** screen which stores information on the development of the swimmers’ skills.

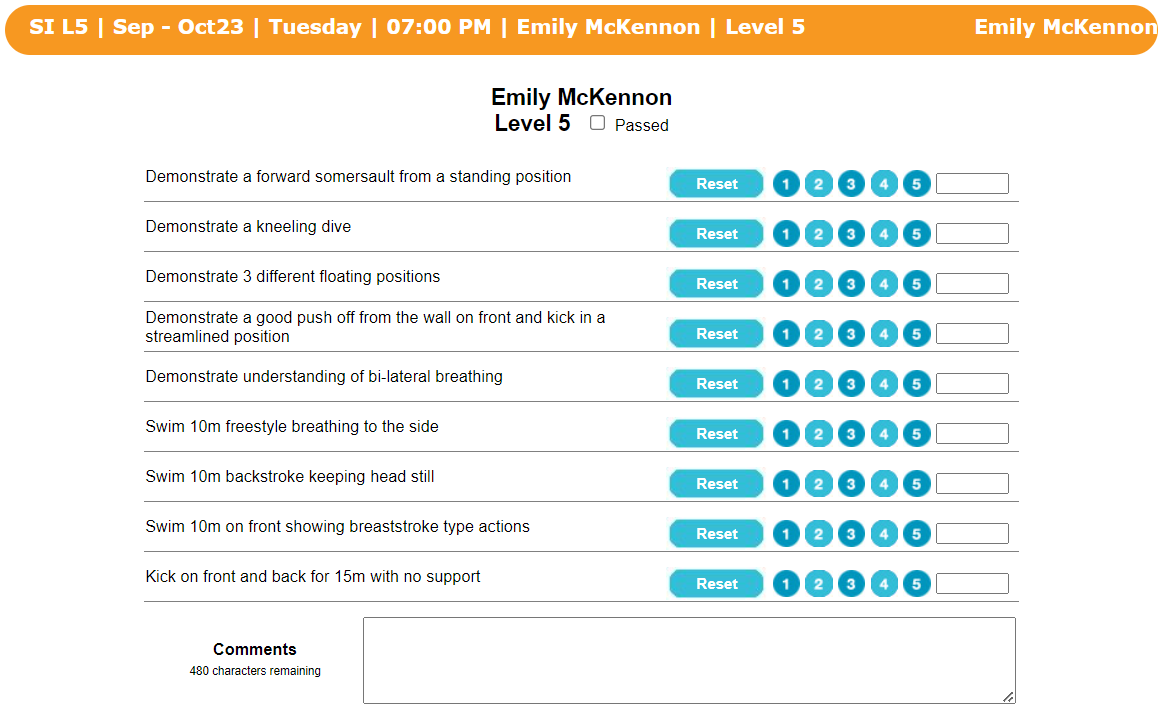
It **shares** the same filter search as the **‘Mark Attendance’** screen and the **Completion colour.**

This has a **‘Old version’** function at the bottom of the screen which **reverts the ‘Progress’** to the previous save. I won’t implement this as I feel **it’s not relevant** to the needs of the project.

A screenshot of a computer

Description automatically generatedThis is the **‘Evaluation’** process for the swimmers. Teachers can **write reviews** on the swimmer’s performance here at the **‘+’** and **move them Up or Down a Level,** they can **Edit/Delete** the review and **‘Add Rewards’.**

This is a better implementation of the ‘**Evaluation Process’** than Morleisure’s solution. I might choose to **implement some of these functions/designs** in my project.

This is the screen where Teachers write the **‘Comment’** for each swimmer.

They can also edit the Marks they give to each swimmer in each **‘Skillset’**

A checkbox to show that the Swimmer has passed the Level.

A screenshot of a computer

Description automatically generated

This is the **‘Move and Stay’** screen where the swimmers are **moved** between each Level.

Again, this screen has **similar functions** to **‘Mark Attendance’ and ‘Student Progress’**.

This links to the **‘Evaluation Process’** screen.

Summary:

Features I would like to implement:

* The **PIN** method will be used instead of the **Username and Password** so it’s easier to store the PIN in the database.
* Functions such as **Filters** for the **Attendance, Progress and Moving swimmers** (as well as each of those functions). A section where Teachers can comment on swimmer’s performance.
* Make **links** between each function i.e. The fact that **‘Mark Attendance’** links with the **‘Registry’** AND how **‘Student Progress’** links with the **‘Evaluation’** process.
* Other **Quality of Life** (QoL) features such as: The **colour** of Registry/Evaluation changing depending on completion, **‘Passed’ checkbox** which sets all skillsets to 5, **‘Current Lesson’** so Teachers don’t have to search for their lesson.

Features I would change:

* The **‘Location and Activities’** screen. This is purely because I will be **only making the functions for the Swimming Classes** and not for Water polo or Summer Classes.
* **Removing the ‘Old Version’** function. Whilst it can be useful to revert to the previous save of ALL information stored on swimmers**, it can be nuisance** as it reverts ALL information, including information **that you might not want changed.**
* **Removing ‘Not Marked’ in Registry.** It’s **confusing to have ‘Not Marked’** box instead of just **Present** or **Absent**. Now there is **only Present and Absent** and the colours associated with **Absent** are changed.

Interview:

I interviewed the manager of the swimming club to gain an insight into the daily struggles on the poolside and to see what changes could be made to improve their current solution:

***Q. “How do you start each of your classes?”***

A. “Well we start by opening our books and going to the allocated time for each of our classes and we call out each of the names on the register. We have a checkbox system for the registration, so we mark them either **Present** or **Absent**. Generally, the kids are late to class, so some teachers must take out the book again in the middle of the lesson and mark them in. It’s quite annoying.

***Q. “How do you find writing down the registry for each of your classes?”***A. “It is tedious of course but quite cost effective! Each registry book lasts us about 1 year, which isn’t the problem. The problem is our teachers spend too much time on marking each of their students and this has caused many parents to complain that they aren’t getting their money’s worth! We need a way to quickly do the registries so that we can teach the classes sooner, this also applies for during lessons.

***Q. “So would a new system benefit your club?”***

A. “100% Yes! A new system would speed up the registry process and make it easier to store them! One of the main problems for our registry solution is that we have no way to store them until the book is filled up to maximise the info stored. In this time, our books can get lost or destroyed due to the humidity of the pool. So, a new system would enable the club to achieve much more.

***Q. “Would you consider a digital system if it was to improve on your data storage?”***

A. “The club have considered a digital lesson manager in the past, but the only problem is the funding for it…if we had a digital system then our work would be much more efficient, we would be able to easily store all our completed registries and keep them secure as well as reduce the costs for the upkeep of data. There is also lots of other information that we could store using this method including swimmer details like emails, special needs etc.

***Q. “You mentioned before that you only store the registers once you fill the books? What would happen if the book was damaged or got lost?”***A. “Well in that situation we would have backups of the registries so it’s not that big of a problem. At the end of each week, we copy down all the registries and notes taken into these separate books we keep in the storage room. This is our makeshift library of swimmer information. Normally we select a few Teachers at the end of the week with the task to copy down all the registries for that week and that just isn’t practical! Some of us need to go home early and especially some of our younger teachers! So, if a book had gotten lost mid-week, it would be impossible to recover it. Thankfully this hasn’t happened yet, but it would be nice to know that it couldn’t happen!

***Q. “Are there any features in this solution that you would like to implement that I haven’t already mentioned?”***

A. “At this point in time our main concern is to get a better lesson manager system up and running for our teachers asap. However, a way to message teachers through the lesson manager software would be very convenient, as normally we would post the Scheme of Work for the week on a WhatsApp group chat and communicate that way. An option to chat through the manager would be a nice touch, but again our priority is to get the lesson manager working first!

Summary Specification:

The purpose of this project is to provide Lisburn LeisurePlex with a new and improved lesson management system. These improvements are as follows:

* A way to show whether swimmers are “Present” of “Absent” and a record of their total attendance that week.
* The ability to store this information locally on the system until it is merged onto the main server.
* A way to check the progress of the swimmer in their level i.e., have a list of criteria the swimmer must meet to pass to the next level.
* A way for the Manager to move swimmers between levels ***and to be notified of a move request for said swimmers.***
* A place for the Manager to post the Scheme of Work so teachers can be aware of what skills to cover that week/***additional chat functions for teachers to communicate with one another.***
* A way for each teacher to see what class they have, the level, their assistant, the time of the class and the names of the swimmers in that class.
* A way for the Manager to view all teachers’ classes and to change information on those classes i.e., Swimmers in class, Teacher, Assistant.

Stakeholders’ Requirements:

Managers:

* Can login to their lesson manager account via the ‘Manager’ login prompt.
* Can view/edit/manage all teachers’ classes as well as assign teachers/assistants to classes.
* Can move swimmers up/down a level and can view the teacher’s notes on each swimmer (the reason that the teacher moved requested them to move up/down).
* Can mark swimmers as either “Present” or “Absent”.
* Can write notes about swimmers.
* Can assess swimmers based on their skills for that level.
* Can put down what equipment is needed for the class (in advance).

Level 2 Teachers:

* Can login to their lesson manager account via the ‘Teacher’ login prompt.
* Can view their classes separately or all together.
* Can write notes about swimmers.
* Can assess swimmers based on their skills for that level.
* Can make requests to move swimmers up/down a level.
* Can mark swimmers as either “Present” or “Absent”.
* Can put down what equipment is needed for the class (in advance).

Level 1 Teachers/Assistants:

* Can login to their lesson manager account via the ‘Teacher’ login prompt.
* Can view what classes they are assisting in and what teacher they are assisting.
* Can mark swimmers as either “Present” or “Absent”.
* Check what equipment is needed for the Lesson through the class information.

**Success Criteria:**

Managers:

* Account Management:
* Can create an account with their PIN.
* Can login to account with their PIN.
* Can change their own PIN.
* Can change other teachers’ PINs.
* Accessing Database:
* Once logged in they can access the current records stored in the Lesson Manager.
* They can merge this data into the main server (stored locally in the club)
* Create Classes:
* Can add swimmers to classes.
* Can assign teachers and assistants to classes.
* Can select level of classes.
* Can set goals for classes (Scheme of Work)
* Create SOW:
* Can put the SOW **for each level** that week here. Manager will have the option to **edit the SOW** showing the **week no., introduction, main, contrast and depth of pool.**
* Managing Classes:
* Can select to view “All lessons”, “My lessons” and “Current lesson”.
* Can edit these classes. This would include **moving swimmers up/down** a level and **adding/removing** swimmers to/from classes.
* Can write notes about classes, including any Special Educational Needs a swimmer may have.
* Can use registries for all classes.
* Can access the syllabus for each class.
* Can assess swimmers.
* Managing Teachers/Assistants:
* Can add teachers/assistants to classes as well as remove them.
* Can promote teachers to managers and assistants to teachers.
* Can view teacher/assistant information. This includes name, age, phone number, email address, PIN number \*and unique worker number\*.
* Can change teacher/assistant PIN numbers.
* Moving Swimmers:
* Can view the ‘Swimmer Details’. This includes **Level, Age, Name and Move Info (**which includes the **reason for the request**)**.**
* Can accept the move **‘Move Request’** for a swimmer. which includes the **reason for the request (‘Move info’).**
* Creating Reports:
* Can write up about the performance of any swimmer, whether they passed the level and any additional advice/information and send it to parents’ emails.
* Can view said reports on the lesson manager. This would display if the swimmer “Passed” or “Failed”, an option to view the report, the teacher who wrote the report, additional information about the swimmer/level.
* Can view all reports written by all teachers. Manager must review reports before sending them to each parent.
* Send Reports:
* A page which the manager views before sending through the report for that class. ***If the manager feels changes need to be made, they can ‘Disapprove’ the report, which notifies the teacher. Else, they ‘Approve’ the report and it’s sent.***

Teachers:

* Account Management:
* Can create an account with their PIN.
* Can login to account with their PIN.
* Managing Classes:
* Can select to view “**All lessons**”, “**My lessons**”, “**Current lesson**” and “**SOW**”.
* Can use the register for each class to mark the attendance for swimmers.
* Can view the syllabus for each class.
* Can assess swimmers in each class.
* Can write notes about each swimmer in the class.
* Create Reports:
* Can create reports on swimmers for the manager to see. Teachers can submit these reports for the manager to review.
* Can **write up** about the **performance** of any swimmer, **whether they passed** the level and any **additional advice/information** and send it to **manager for review**.
* Can view all reports that they wrote and can edit them.
* Send Reports:
* Can send reports to the manager for review. If the report is ‘**Disapproved’** then the **teacher is notified** and the **report for that class is highlighted.**
* Moving Swimmers:
* Can **create a ‘Move Request’** which contains ‘**Swimmer Details’** and the **‘Move info’.**

Assistant:

* Account management:
* Can create an account with their PIN.
* Can login to account with their PIN.
* Managing classes:
* Can select to view “**All my lessons**”, “**Current lesson**” and “**SOW**”.
* Can use the register for each class to mark the attendance for swimmers.
* Can view the syllabus for each class.
* Can assess swimmers in each class.