|  |
| --- |
| Non-Examined Assessment |
| Dayplan Creator |

William E Martin (3605)

Godalming College (64395)

Table of Contents

[Table of Contents 2](#_Toc478643001)

[Research 3](#_Toc478643002)

[End user Overview 3](#_Toc478643003)

[The Problem 3](#_Toc478643004)

[Interview with end user 4](#_Toc478643005)

[Question – Answer summery 4](#_Toc478643006)

[Analysis 8](#_Toc478643007)

[Project Requirements 8](#_Toc478643008)

[Data flow diagrams 9](#_Toc478643009)

[Current system flow table 10](#_Toc478643010)

[Data dictionary 10](#_Toc478643011)

[Lifeguard Table 10](#_Toc478643012)

[Timetable Table 11](#_Toc478643013)

[Rota Timetable 11](#_Toc478643014)

[Design 17](#_Toc478643015)

[User interface 18](#_Toc478643016)

[19](#_Toc478643017)

[Solution Path 19](#_Toc478643018)

[System Requirements 19](#_Toc478643019)

[Templates 20](#_Toc478643020)

[Class Diagrams 23](#_Toc478643021)

[Pseudo Code 23](#_Toc478643022)

[Quicksort 23](#_Toc478643023)

[DDL Statements 24](#_Toc478643024)

[SQL Statements 24](#_Toc478643025)

[Validation 24](#_Toc478643026)

[Technical Solution – Explained 24](#_Toc478643027)

[Security 24](#_Toc478643028)

[Testing 25](#_Toc478643029)

[Evaluation 25](#_Toc478643030)

[User feedback 25](#_Toc478643031)

[Personal Evaluation 25](#_Toc478643032)

[Possible extensions 25](#_Toc478643033)

[Web page 25](#_Toc478643034)

[Excel Formatting 25](#_Toc478643035)

[Evaluating requirements 26](#_Toc478643036)

[Project Conclusion 26](#_Toc478643037)

[Appendix 1 - Source Code 27](#_Toc478643038)

# Research

# End user Overview

End User Overview

Name: Sam Horne

Organisation: Places for People Leisure

Village Way,

Cranleigh,

Surrey,

GU6 8AF

Contact:

[samhorne@pfpleisure.org](mailto:samhorne@pfpleisure.org)

<http://www.placesforpeopleleisure.org/centres/cranleigh-leisure-centre/>

01483 274400

The end user, Sam Horne, is a Manager at the Cranleigh leisure Centre in Cranleigh. The sub-users will be Duty Managers. Duty Managers have to plan out the day of the Lifeguards that are working each day to make sure that everyone has a job and that the pool is constantly under observation.

# The Problem

Planning these days out can be time consuming, especially if the dayplan wants to be even close to optimal. This wastes the Managers time as they can be called off to more immediate problems throughout the day. A new system greatly reducing the time needed to make these plans in needed. Currently there are no alternatives online where a customisable timetabling system can be bought.

# Interview with end user

I have interviewed the main user to directly gather as much information about the current system and to find out what is needed to solve the current problem. By doing an interview I can make sure that I find out what I need by asking specific questions as well as getting more detailed answers in specific areas rather than just a general questionnaire that might not even get the information that I need. This will give me more of an accurate view as to how to go about producing a solution.

I am interviewing Sam Horne, because as a general manager he has more experience with this problem and has had to cater to more complex scenarios in larger leisure centres

# Question – Answer summery

Question

Reason for question

Answer

**What is the current system for Timetabling?**

This allows for me to develop my understanding for how the current system works and how well it performs

People sign up for shifts in the work rota folder. Day plans are then created every night for the following day by the duty manager. The work rota is checked to see who is working the next day and from what time. This is all done with the use of Microsoft Excel.

**How do you decide if a full-timer works an early or a late?**

Need to understand any constrains in system

All workers must have at least 11 hours between each shift meaning if they work a late shift that finishes at 10 pm they cannot work until 9 am at the earliest the following day.

**How useful would a new system be? (Computer application**

A new system sounds very interesting and useful as the current system is flawed, time consuming and regularly has errors that are missed.

**What flaws are there current system are there?**

To identify where improvements and fixes can be made in the current system

As this is all done by hand people could and have in the past altered names in the sign-up rota either because they want a shift that is taken or they no longer want to work a shift that they signed up for. Once you sign up for a shift you are required to give 2 weeks’ notice if you can no longer work it at any point.

The day plan also can take longer for people to do with less experience and can be hard to ‘optimise’ and give people minimum time poolside. Sometimes people make mistakes when making the day plan and don’t put someone on poolside when it is needed and we have to ‘improvise’ and drag on someone who is on a clean.

**How would you like these flaws addressed?**

To find if there would be preferred method to tackle a specific problem

If the ‘human element’ of creating the day plan could be removed to stop day plan errors that would be great. Beyond that a way of stopping people from altering shifts in the sign-up rota would be helpful.

**Constraints of timetabling?**

To find out any check that would need to be made within the programme

* Length of each shift-change

Each 30 mins people will swap their current task (either main pool, teach pool, clean or lunch)

* Maximum work time

All people can work a maximum of 48 hours a week unless they ‘opt out’ and under 18’s can work a maximum of an 8 hour shift and cannot work after 10 pm.

* Lunch breaks

Under 18’s get half an hour lunch break after 5 hours work. Over 18’s must work 8 hours for a lunch break and 9 hours or more for 2 half hour lunch breaks.

* Number of people needed at each time

There needs to be a minimum of 2 – 4 people depending on what the time is and what is happening on poolside but ideally there is always 1 more lifeguard than the number of people needed on poolside at any one time.

* Type of activities

Lane swim, general swim, disabled swim, over 50’s swim, all welcome (Programmed and Un-Programmed)

* Type of task on timetables

Main pool / deep, Main pool / shallow, Main pool, Teaching pool, Clean, Lunch

**How would you like this system to work?**

To find out what features would be liked

Ideally the new sign-up system would not allow people to alter their shift without talking to a duty manager or giving two weeks’ notice.

If the new system could automatically create a day plan by looking at 1) the people working each day and 2) the number of people needed on poolside based off of the poolside activities.

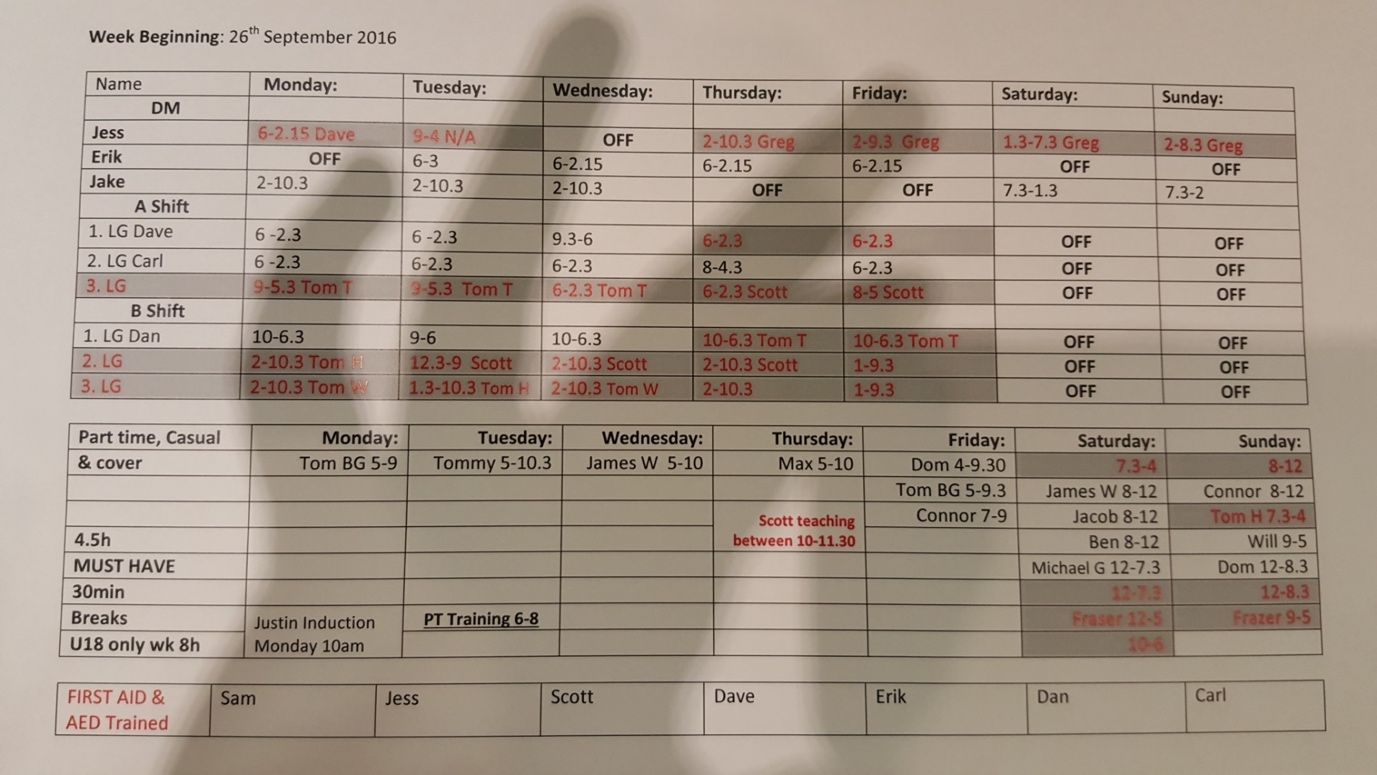
Duty managers would also need access to change the sign-up rota at short notice in case someone say phones in sick.

Observation

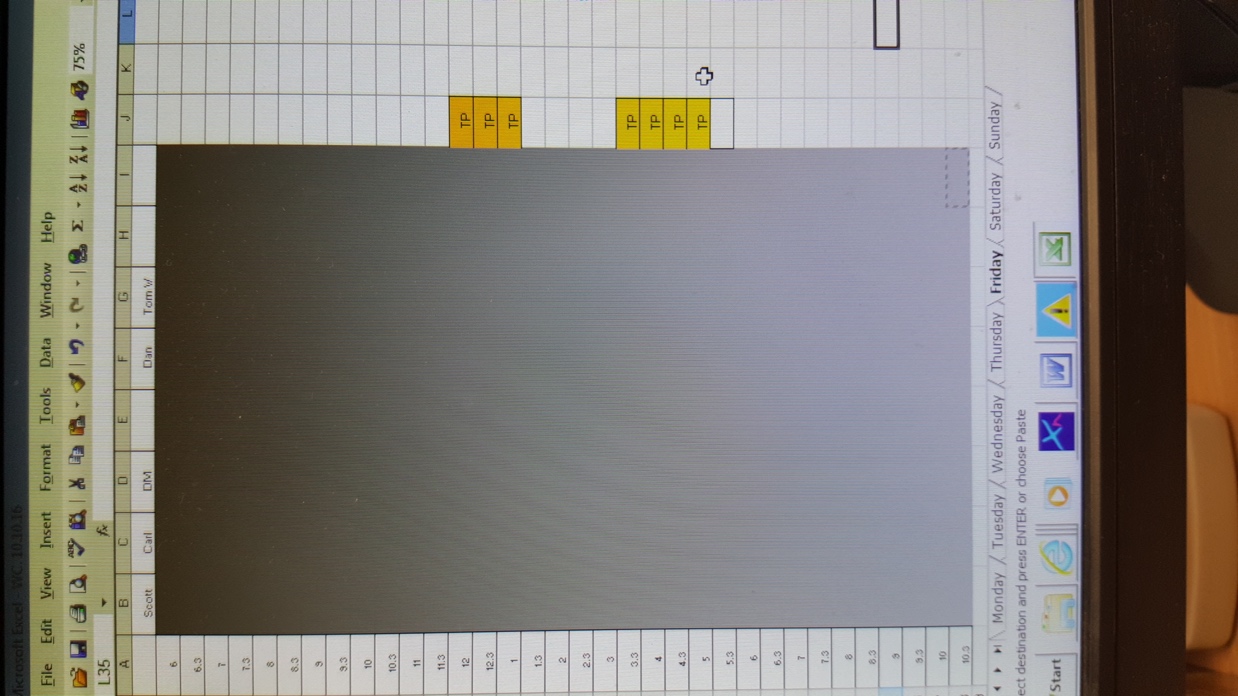
To sign up for work people write their name in a box allocated to a shift that they want to work in the sign-up folder. The folder is checked every day and all names written in are added on the excel spreadsheet and is then printed off and placed back into the folder so that the names can’t be changed.

The day plan is written every night for the following day by the Duty Manager who is working the late-shift. The sign-up folder is checked to see who is working and at what time the following day. A day plan is then created from a template on excel which has half hour slots. Every half hour there needs to be the correct number of people allocated on poolside. This ranges between 1 and 3 lifeguards. [MP/D] for main pool deep & [MP/S] for main pool shallow (this is when two people are needed on main pool), [MP] for main pool (when one person is needed on main pool) and [TP] for teaching pool. Every lifeguard not on poolside is on a cleaning shift [C] or their lunch [L].

The day plan is made on excel and has to be checked for errors. They check that there are always enough people on poolside every half hour, that someone has not been on poolside for more than an hour and a half without a half hour break. They also check that people have the correct number of lunches allocated to them in their shift and that there is always at least one lifeguard on a clean in case there is an emergency. Once the day plan is checked it is then printed off for the next day.



Current Rota layout



Current Template and example dayplan

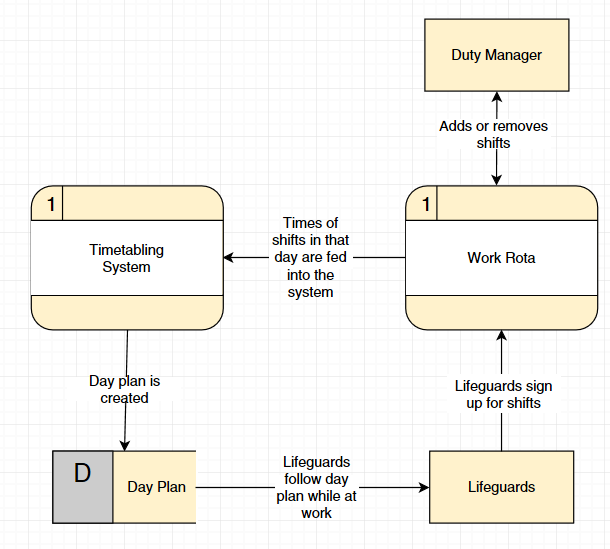
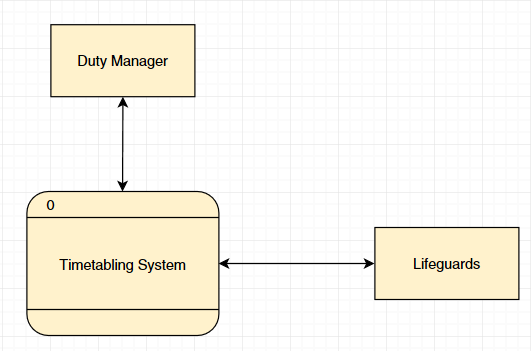
# Analysis

My investigation has shown how the current timetabling system works and the flaws that need to be addressed. I have determined that the best solution to this is to have an online version of the current work rota. At the end of each day the day plan will be created and printed off using the information in the work rota. Lifeguards and Duty managers will also be able to view the work rota. The end user has also requested an admin login to allow control of the work rota.

# Project Requirements

1. To be able to edit work rota
   1. To be able to add shifts to the database
   2. To be able to edit current shifts in the database
   3. To be able to delete shifts in the database
2. To be able to edit lifeguard registry
   1. To be able to add lifeguards to register
   2. To be able to edit lifeguard information on the register
   3. The be able to remove lifeguards from the register
3. To be able to save dayplan requirements
   1. To be able to save and load dayplan requirements to and from a specified text document
4. To be able to view the Dayplan after creation
   1. Must be in an understandable format to allow the user to view the dayplan
   2. Should be in a format which allows the user to print the dayplan
   3. le table format
5. Must be an easy to understand interface
   1. Lifeguards and duty managers should not struggle to use the interface

# Data flow diagrams



# Current system flow table

|  |  |
| --- | --- |
| INPUT   * Lifeguards working on that day * Shifts of the lifeguards * Pool activities throughout the day | PROCESS/DECISION   * Open Template * Fill template * Check meets constraints |
| STORAGE   * Temporary save till end of week | OUTPUT   * Print Dayplan |

# Data dictionary

All information needed to create the Dayplan will need to be sorted in a database for ease of general access as well as any additions or edits that might need to be made in the future. Hence I have created a data dictionary for the database that I will make to store this information

# Lifeguard Table

| Name | Purpose | Data Type | Data Size | Example Data | Validation |
| --- | --- | --- | --- | --- | --- |
| Payroll  Number | To store the unique ID given to each lifeguard when created | Integer | 6  (default) | 123456 | A natural number that has a character length 6 |
| First Name | To store the name of the lifeguard | String | 30 | Henry | A non-numeric string no more than 30 characters long |
| Surname | To store the name of the lifeguard | String | 30 | Smith | A non-numeric string no more than 30 characters long |
| Age | To store the age of the lifeguard | Integer | 2 | 20 | A natural number less than 100 but at least 15 |

### Timetable Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Purpose | Data Type | Data Size | Example Data | Validation |
| Payroll Number | To link a lifeguard with a specific shift | Integer | 6  (default) | 123456 | A natural number that has a character length 6 |
| ShiftID | To link a shift with the lifeguard working that shift | string | 6  (default) | 12MIDL | A String formed of the start time and shift length eg MIDL, middling length |

# Rota Timetable

| Name | Purpose | Data Type | Data Size | Example Data | Validation |
| --- | --- | --- | --- | --- | --- |
| ShiftID | To link a shift with the lifeguard working that shift | string | 6  (default) | 12MIDL | A String formed of the start time and shift length eg MIDL, middling length |
| Day | To store the day that the shift takes place | String | 9 | Wednesday | N/A |
| Date | To store the date that the shift takes place | String | String | 25 | 01/01/2017 |
| Shift Start | To store the start time of the shift | Decimal | 3 | 12.5  (12:30pm) | Decimal between 6.0 and 22.5 (earliest opening and latest closing hours) |
| Shift End | To store the end time of the shift | Decimal | 3 | 12.5  (12:30pm) | Decimal between 6.0 and 22.5 (earliest opening and latest closing hours) |

Document Specification Sheet -

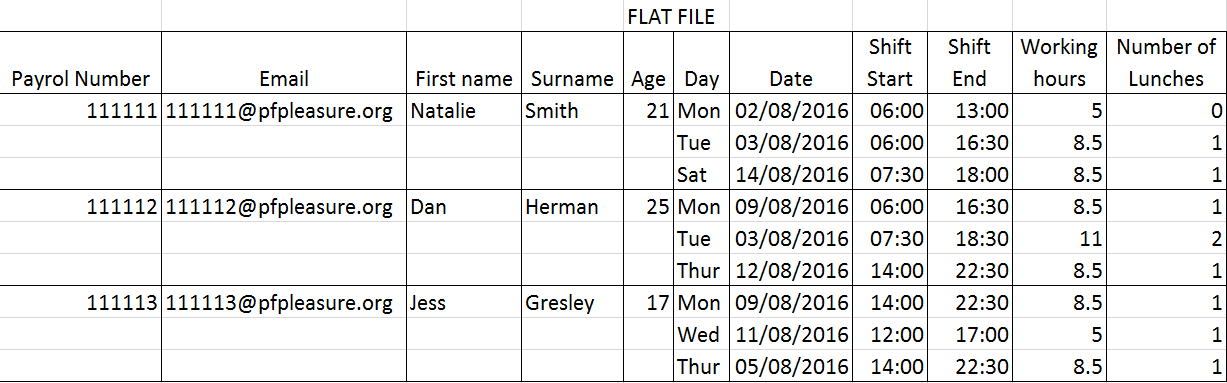
Current System

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Volumetrics** | | | | | | | | | | | |
| Document description | | | System | | | | Document | | Name | | Sheet |
| Work Rota | | | Timetabling | | | | 1 | | Will Martin | | 1 |
| Stationery ref. | | | Size | | | | Number of parts | | Method of preparation | | |
|  | | | A4 | | | | 1 | | Typed in Microsoft excel | | |
| Filing sequence | | | | | Medium | | | | Prepared by | | |
|  | | | | | Paper Document | | | | Duty Manager | | |
| Frequency of preparation | | | | | Retention period | | | | Location of file | | |
| Once a week | | | | | 6 weeks | | | | Work Folder in Duty Managers office | | |
| Volume | Minimum | | | Maximum | | Av/Abs | | Growth rate/fluctuations | | | |
| 1 per week | | | 7 per week | | 7 per week | | none | | | |
| Users/receipts | | Purpose | | | | | | | | Frequency of use | |
| Duty Manager, Lifeguards | | Maintaining a record of when anyone lifeguard or duty manager is working during the next six weeks | | | | | | | | ∞ | |
| **Data Dictionary** | | | | | | | | | | | |
| Ref | Name | Data Type | | | Regex | | | Occurrence | | Source of data / description | |
| 1 | DM1 | A | | |  | | | 6 | | Duty Manager | |
| 2 | DM2 | A | | |  | | | 6 | | Duty Manager | |
| 3 | LG1 – fulltime | A | | |  | | | 6 | | Duty Manager | |
| 4 | LG2 – fulltime | A | | |  | | | 6 | | Duty Manager | |
| 5 | LG3 – fulltime | A | | |  | | | 6 | | Duty Manager | |
| 6 | LG4 – fulltime | A | | |  | | | 6 | | Duty Manager | |
| 7 | LG5 – part-time | A | | |  | | | 6 | | Lifeguards | |
| 8 | LG6 - part-time | A | | |  | | | 6 | | Lifeguards | |
| 9 | Day of week | A | | |  | | | 42 | | Duty Manager | |
| 10 | 08.00 – 12.00 | A, 9 | | |  | | | 5 | | Duty Manager, Lifeguards | |
| 11 | 07.30 – 16.00 | A, 9 | | |  | | |  | | Duty Manager, Lifeguards | |
| 12 | 09.00 – 17.00 | A, 9 | | |  | | | 1 | | Duty Manager, Lifeguards | |
| 13 | 12.00 – 17.00 | A, 9 | | |  | | | 4 | | Duty Manager, Lifeguards | |
| 14 | 12.00 – 20.30 | A, 9 | | |  | | | 2 | | Duty Manager, Lifeguards | |
| 15 | 12.00 – 19.30 | A, 9 | | |  | | | 1 | | Duty Manager, Lifeguards | |
| 16 | 14.00 – 10.30 | A, 9 | | |  | | | 10 | | Duty Manager, Lifeguards | |
| 17 | 14.00 – 21.00 | A, 9 | | |  | | | 6 | | Duty Manager, Lifeguards | |
| 18 | 17.00 – 22.00 | A, 9 | | |  | | | 8 | | Duty Manager, Lifeguards | |
| 19 | 06.30 – 14.00 | A, 9 | | |  | | | 10 | | Duty Manager, Lifeguards | |

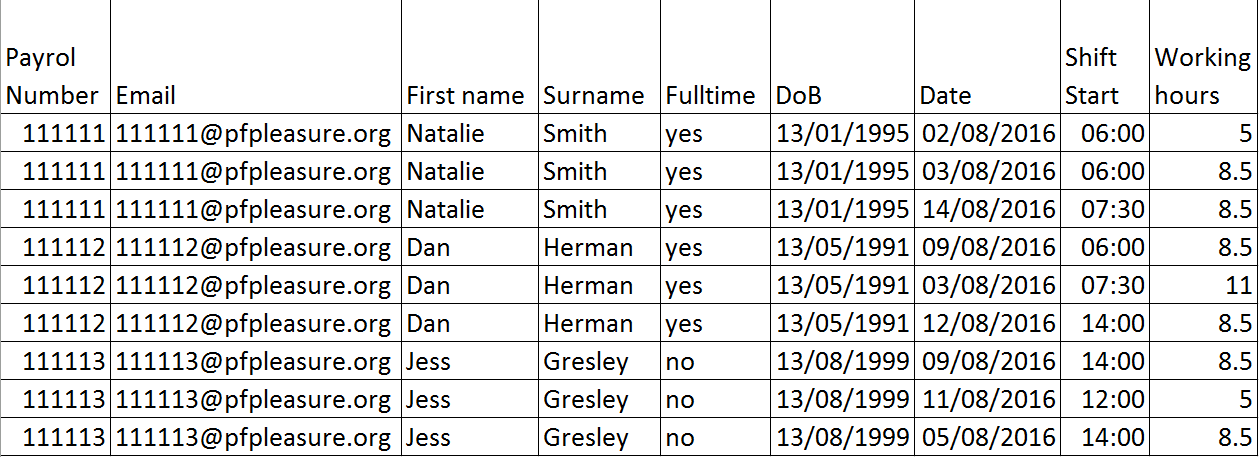
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Volumetrics** | | | | | | | | | | | |
| Document description | | | System | | | | Document | | Name | | Sheet |
| Day Plan | | | Timetabling | | | | 1 | | Will Martin | |  |
| Stationery ref. | | | Size | | | | Number of parts | | Method of preparation | | |
|  | | | A4 | | | | 1 | | Typed in Microsoft Excel | | |
| Filing sequence | | | | | Medium | | | | Prepared by | | |
| 30 – min slots | | | | | Paper Document | | | | Duty Manager | | |
| Frequency of preparation | | | | | Retention period | | | | Location of file | | |
| 1 per day | | | | | 7 days | | | | Staffroom and duty managers office | | |
| Volume | Minimum | | | Maximum | | Av/Abs | | Growth rate/fluctuations | | | |
| 1 | | | 2 | | 1 | | none | | | |
| Users/receipts | | Purpose | | | | | | | | Frequency of use | |
| Duty Manager, lifeguards | |  | | | | | | | |  | |
| **Data Dictionary** | | | | | | | | | | | |
| Ref | Name | Data Type | | | Regex | | | Occurrence | | Source of data / description | |
| 1 | 06.00 - 06.30 | 9,A | | |  | | | 1 | | Duty Manager | |
| 2 | 06.30 - 07.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 3 | 07.00 - 07.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 4 | 07.30 - 08.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 5 | 08.00 - 08.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 6 | 08.30 - 09.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 7 | 09.00 - 09.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 8 | 09.30 - 10.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 9 | 10.00 - 10.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 10 | 10.30 - 11.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 11 | 11.00 - 11.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 12 | 11.30 - 12.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 13 | 12.00 - 12.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 14 | 12.30 - 13.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 15 | 13.30 - 14.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 16 | 14.00 - 14.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 17 | 14.30 - 15.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 18 | 15.00 - 15.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 19 | 15.30 - 16.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 20 | 16.00 - 16.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 21 | 16.30 - 17.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 22 | 17.00 - 17.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 23 | 17.30 - 18.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 24 | 18.00 - 18.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 25 | 18.30 - 19.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 26 | 19.00 - 19.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 27 | 19.30 - 20.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 28 | 20.00 - 20.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 29 | 20.30 - 21.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 30 | 21.00 - 21.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 31 | 21.30 - 22.00 | 9 | | |  | | | 1 | | Duty Manager | |
| 32 | 22.00 - 22.30 | 9 | | |  | | | 1 | | Duty Manager | |
| 33 | MP | A | | |  | | | Variable | | Duty Manager | |
| 34 | MP/S | A | | |  | | | Variable | | Duty Manager | |
| 35 | MP/D | A | | |  | | | Variable | | Duty Manager | |
| 36 | TP | A | | |  | | | Variable | | Duty Manager | |
| 37 | C | A | | |  | | | Variable | | Duty Manager | |
| 38 | L | A | | |  | | | Variable | | Duty Manager | |

# Design

Normalization

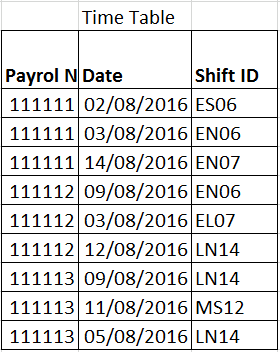
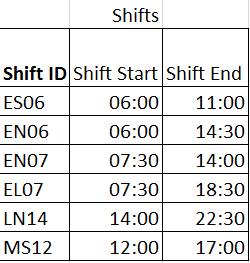


This flat-file is immediately redundant as there are repeats of data within the table. The shift end attribute is redundant as it can be calculated from the ‘working hours’ attribute by adding it onto the start time to work out the end time. The day attribute can be calculated from the date by a person and is redundant information so can be removed. The part time attribute is redundant as if someone is not fulltime then they must be part time. Age is needed but this is something that would need to be updated every year therefore it is more sensible to use ‘Date of Birth’ instead. Number of lunches is also not needed as lunches are dependent on ‘Working Hours’ and ‘Age’. By making these alterations the database is now in 1NF.



Once in 1NF to change it to 2NF we need different tables to eliminate data repetitions like email repeats each table is linked to others via primary and compound keys



Once the tables are linked by Keys they form and ‘Entity relationship’ table the database is now in 3NF as the data in each table relies on the key, the whole key and nothing but the key. Unfortunately due to the function of this database the date is required to be stored as people need to know what date they are working. This will unfortunately create lots of entries in the ‘time table’ table, one for each shift a person ever works. This could be reduced in use by archiving the data in the ‘time table’ table every 3 months.

# User interface

The system will be accessible in one place allowing the creation of the day plan to be quick and easy. There will be a method of entering, saving and loading the required lifeguards in the interface as well and a way to add, edit and remove, shifts or lifeguards to and from the database.

There will also be a way to create and then view the Dayplan immediately in a printable format to allow the end user to get the job done.

VB forms user interface

# https://i.gyazo.com/14ec39b2f84e33a1da311285641e3147.png

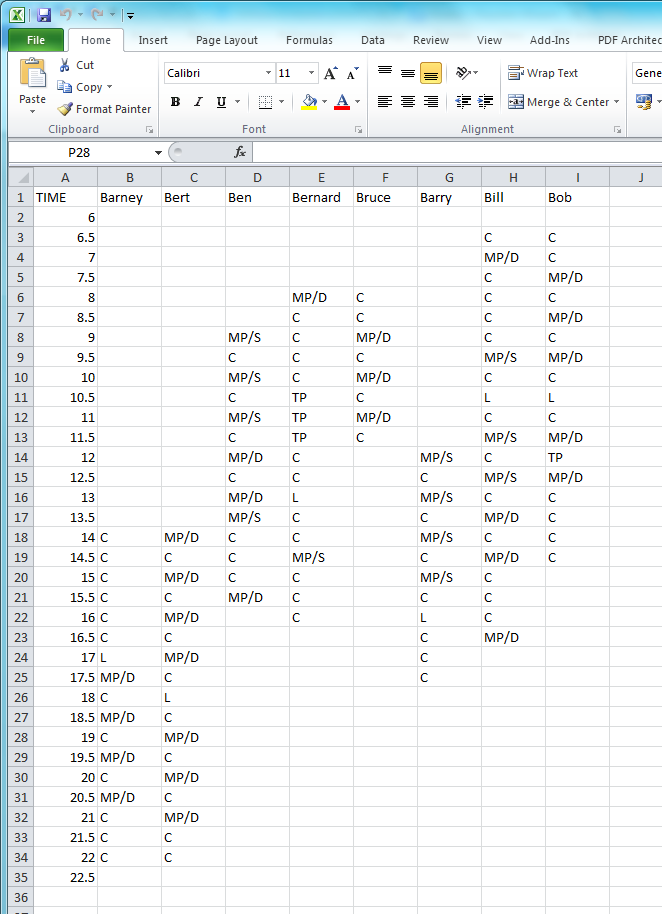
# Solution Path

To create my solution to this project I will concentrate coding the Dayplan creation in a way that will optimise the plan to reduce the time that someone is on poolside to a fair amount and test it with hardcoded data before linking the project to a database which will hold test data and be able to add, edit, remove more information. Once I have debugged the code I can add a user interface with VB window forms to allow the end user to use the program in an easy and understandable manner.

# System Requirements

The main system will be in VB Forms, an easily installed and free program which will allow the creation and optimisation of a Dayplan. The system is in two parts, a server with MySQL installed run through XAMPP where all the lifeguard and shift records will be stored and accessed by the project. The second half of the system will be the VB forms which will allow the user to interact with the program and to create the Dayplan.

The system depends on the database which can be hosted on any server but is more easily done on a local computer with XAMPP installed.



Example dayplan created from the output of the VB solution

# Templates

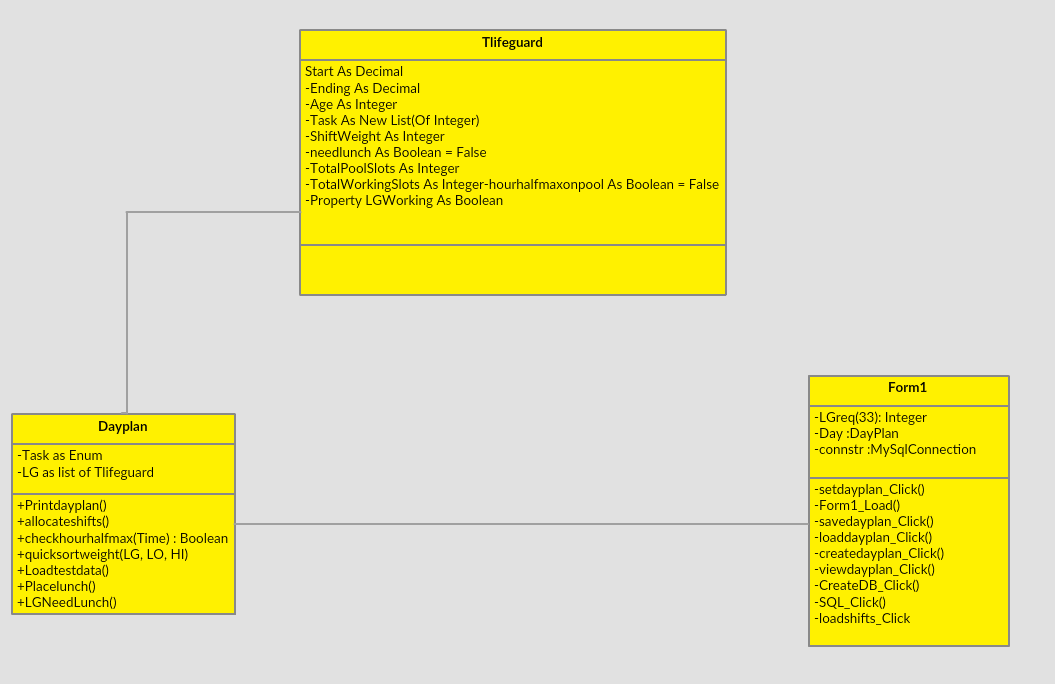
Template of output (day plan)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | LG1 | LG2 | LG3 | LG4 | LG1 | LG2 | LG3 | LG4 |
| Time |  |  |  |  |  |  |  |  |
| 06.00 - 06.30 |  |  |  |  |  |  |  |  |
| 06.30 - 07.00 |  |  |  |  |  |  |  |  |
| 07.00 - 07.30 |  |  |  |  |  |  |  |  |
| 07.30 - 08.00 |  |  |  |  |  |  |  |  |
| 08.00 - 08.30 |  |  |  |  |  |  |  |  |
| 08.30 - 09.00 |  |  |  |  |  |  |  |  |
| 09.00 - 09.30 |  |  |  |  |  |  |  |  |
| 09.30 - 10.00 |  |  |  |  |  |  |  |  |
| 10.00 - 10.30 |  |  |  |  |  |  |  |  |
| 10.30 - 11.00 |  |  |  |  |  |  |  |  |
| 11.00 - 11.30 |  |  |  |  |  |  |  |  |
| 11.30 - 12.00 |  |  |  |  |  |  |  |  |
| 12.00 - 12.30 |  |  |  |  |  |  |  |  |
| 12.30 - 13.00 |  |  |  |  |  |  |  |  |
| 13.30 - 14.00 |  |  |  |  |  |  |  |  |
| 14.00 - 14.30 |  |  |  |  |  |  |  |  |
| 14.30 - 15.00 |  |  |  |  |  |  |  |  |
| 15.00 - 15.30 |  |  |  |  |  |  |  |  |
| 15.30 - 16.00 |  |  |  |  |  |  |  |  |
| 16.00 - 16.30 |  |  |  |  |  |  |  |  |
| 16.30 - 17.00 |  |  |  |  |  |  |  |  |
| 17.00 - 17.30 |  |  |  |  |  |  |  |  |
| 17.30 - 18.00 |  |  |  |  |  |  |  |  |
| 18.00 - 18.30 |  |  |  |  |  |  |  |  |
| 18.30 - 19.00 |  |  |  |  |  |  |  |  |
| 19.00 - 19.30 |  |  |  |  |  |  |  |  |
| 19.30 - 20.00 |  |  |  |  |  |  |  |  |
| 20.00 - 20.30 |  |  |  |  |  |  |  |  |
| 20.30 - 21.00 |  |  |  |  |  |  |  |  |
| 21.00 - 21.30 |  |  |  |  |  |  |  |  |
| 21.30 - 22.00 |  |  |  |  |  |  |  |  |
| 22.00 - 22.30 |  |  |  |  |  |  |  |  |

Example of Finished output

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | LG1 | LG2 | LG3 | LG4 | LG1 | LG2 | LG3 | LG4 |
| Time | Sarah | Will | Dave |  | Tom T | Scott | Dan | Erik |
| 06.00 - 06.30 | C | C |  |  |  |  |  |  |
| 06.30 - 07.00 | MP | C |  |  |  |  |  |  |
| 07.00 - 07.30 | C | MP |  |  |  |  |  |  |
| 07.30 - 08.00 | MP | C |  |  |  |  |  |  |
| 08.00 - 08.30 | C | MP |  |  |  |  |  |  |
| 08.30 - 09.00 | MP | C |  |  |  |  |  |  |
| 09.00 - 09.30 | C | C | MP |  |  |  |  |  |
| 09.30 - 10.00 | C | MP | C |  |  |  |  |  |
| 10.00 - 10.30 | MP | C | C |  | C |  |  |  |
| 10.30 - 11.00 | C | C | C |  | MP |  |  |  |
| 11.00 - 11.30 | C | C | MP |  | C |  |  |  |
| 11.30 - 12.00 | L | MP | C |  | C |  |  |  |
| 12.00 - 12.30 | MP | L | C |  | TP |  |  |  |
| 12.30 - 13.00 | C | TP | MP |  | C |  |  |  |
| 13.30 - 14.00 | MP | C | L |  | TP |  |  |  |
| 14.00 - 14.30 | C | C | C |  | C | MP | TP |  |
| 14.30 - 15.00 |  |  | MP |  | TP | C | C |  |
| 15.00 - 15.30 |  |  | C |  | L | TP | MP |  |
| 15.30 - 16.00 |  |  | TP |  | MP/D | MP/S | C |  |
| 16.00 - 16.30 |  |  | C |  | MP/S | TP | MP/D |  |
| 16.30 - 17.00 |  |  | MP/D |  | TP | C | MP/S |  |
| 17.00 - 17.30 |  |  | MP/S |  | C | MP/D | C | TP |
| 17.30 - 18.00 |  |  |  |  | MP/D | C | TP | MP/S |
| 18.00 - 18.30 |  |  |  |  | C | MP | L | C |
| 18.30 - 19.00 |  |  |  |  |  | L | C | MP |
| 19.00 - 19.30 |  |  |  |  |  | C | MP | L |
| 19.30 - 20.00 |  |  |  |  |  | MP | C | C |
| 20.00 - 20.30 |  |  |  |  |  | C | C | MP |
| 20.30 - 21.00 |  |  |  |  |  | C | MP | C |
| 21.00 - 21.30 |  |  |  |  |  | MP | C |  |
| 21.30 - 22.00 |  |  |  |  |  | C | MP |  |
| 22.00 - 22.30 |  |  |  |  |  | C | C |  |

# Class Diagrams



# Pseudo Code

# Quicksort

Sub quicksort(numbers, LO, HI)

x ← LO

y ← HI

mid ← numbers(Int((x + y) / 2))

Do While x <= y

Do While numbers(x) < mid

x ← x + 1

Loop

Do While mid < numbers(y)

y ← y - 1

Loop

If x <= y Then

swap numbers

x ← x + 1

y ← y - 1

End If

Loop

If LO < y Then Call quicksort(numbers, LO, y)

If HI > x Then Call quicksort(numbers, x, HI)

End Sub

Sub

# DDL Statements

CREATE DATABASE lifeguards

CREATE TABLE IF NOT EXISTS `lifeguards`.`Timetable` ( `ShiftID` INT(5) NOT NULL ,`Payrol Number` INT(6) NOT NULL ) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `lifeguards`.`Rota` ( `ShiftID` INT(5) NOT NULL , `Day` TEXTNOT NULL , `Date` DATE NOT NULL , `Shift Start` DECIMAL(3,1) NOT NULL , `Shift End` DECIMAL(3,1) NOT NULL ) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `lifeguards`.`Lifeguards` ( `Payrol Number` INT(6) NOT NULL ,`First Name` TEXT NOT NULL , `Surname` TEXT NOT NULL , `Age` INT(2) NOT NULL )ENGINE = InnoDB;

## SQL Statements

INSERT INTO `Table` (`Column1`, `Column2`, `Column3`, `Column4`) VALUES (‘Value1’, 'Value2', ' Value3', ' Value4’);

UPDATE `Table` SET (`Column1`=`NewValue`) WHERE (`Column1`=`Value`);

DELETE FROM `Table` Where `Column`=`value`

SELECT lifeguard.`First Name`, lifeguard.age, rota.`start time`, rota.`end time`

FROM ((timetable

INNER JOIN lifeguard ON timetable.`Payroll Number` = lifeguard.`Payroll Number`

INNER JOIN rota ON timetable.`ShiftID` = rota.`ShiftID` WHERE rota.date = aDate

# Validation

In order for the database information to be entered correctly I have validated the entry’s within the programme

# Technical Solution – Explained

The system lets the end user create a Dayplan using information pulled from the database which can then be easily viewed and printed as well an interface to edit the information on the database. My Project is mainly written in VB with some SQL statements, a popular and free choice, for the database. My code is written in an Object Orientated Programming style to make the project clear to edit.

## Security

As the program does not store any sensitive data, I do not see the need for extra measures such as encryption or two step verification. However the database and programme should still be secured using a password protected login on the computers operating by the main database user of the system that the database is running. Data should also be backed up regularly to prevent loss of information.

# Testing

These are videos evidencing that my project works showing it running.

<https://www.youtube.com/watch?v=SdarhsVnDNQ>

<https://www.youtube.com/watch?v=0DFH7QPYMSo>

<https://www.youtube.com/watch?v=UkVdkyboIbU>

<https://www.youtube.com/watch?v=sD-K61vZY3Q>

# Evaluation

# User feedback

I was in contact with the end user, Sam Horne, throughout the creation of my project allowing me to keep him updated on the progress and to show him thing like the user interface and to let him suggest features that he would like implemented throughout and to let me focus on the things he deemed more important features. I presented my system to Sam and he was very pleased with the results as this would drastically cut down the time that it would take to create the dayplan and found it easy to understand when I explained how to use the programme.

# Personal Evaluation

I found developing this software a long and challenging process but allowed me to develop and improve my understanding of VB forms and allowed me to vastly improve my working knowledge of SQL and DDL.

However I have not finished the project in its entirety as ideally I would create a webpage which would allow other lifeguards to sign themselves up to shifts. The database however is fully integrated with the forms allowing you to add, edit and delete to and from the database as well as select information from the database and to use that in the creation of the dayplan.

# Possible extensions

# Web page

I could add a webpage with admin and lifeguard logins where you could sign up for and view shifts that you were signed up for. The page could then be linked to the database to add shifts that people signed up for to the database.

# Excel Formatting

If I could format the excel spreadsheet output so that different activates where coloured like in my example templates (above) to improve the readability of the dayplan however I could not find the references to download in time to allow me to specify the spreadsheet formatting.

# Evaluating requirements

|  |  |
| --- | --- |
| To be able to edit work rota | This can be done with sql via the forms interface |
| To be able to add shifts to the database | This can be done with sql via the forms interface |
| To be able to edit current shifts in the database | This can be done with sql via the forms interface |
| To be able to delete shifts in the database | This can be done with sql via the forms interface |
| To be able to edit lifeguard registry | This can be done with sql via the forms interface |
| To be able to add lifeguards to register | This can be done with sql via the forms interface |
| To be able to edit lifeguard information on the register | This can be done with sql via the forms interface |
| The be able to remove lifeguards from the register | This can be done with sql via the forms interface |
| To be able to save and load dayplan requirements to and from a specified text document | This can be done by entering the desired file name in the text box adjacent to the save and load buttons |
| To be able to view the Dayplan after creation | This is done by pressing the view dayplan button which presents it as an excel document |
| Must be in an understandable format to allow the user to view the dayplan | Presented as an excel spreadsheet not with much formatting |
| Should be in a format which allows the user to print the dayplan | It can be printed off the excel document |
| Must be an easy to understand interface | Very self-explanatory interface and easy to use |

# Project Conclusion

It is clear from Sam that my project when finished would be extremely useful to the leisure centre. There are some changes that could be made to improve usability and to extend the project allowing for greater system abilities that would be useful.

# Appendix 1 - Source Code

Imports System.IO

Imports System.Diagnostics

Imports System.Data

Imports MySql.Data

Imports MySql.Data.MySqlClient

Public Class Form1

Dim LGreq(33) As Integer

Dim Day As New DayPlan

Dim connstr As MySqlConnection

Function getLGreq() As Integer()

Return LGreq

End Function

Private Sub setdayplan\_Click(sender As System.Object, e As System.EventArgs) Handles setdayplan.Click

Dim hourint, hourindex As Integer

Dim x, hourdbl As Decimal

Dim OutputString, hourval As String

hourval = lstTimes.SelectedItem

hourdbl = CDbl(hourval)

hourdbl = hourdbl \* 2

hourint = CInt(hourdbl)

hourindex = hourint - 12

Try

If LGrequirement.Text < 4 And LGrequirement.Text >= 0 Then

LGreq(hourindex) = LGrequirement.Text

Else

MsgBox("enter a whole number between 0 and 3")

End If

Catch ex As Exception

MsgBox("enter a whole number between 0 and 3")

End Try

Displayreq.Items.Clear()

For x = 6 To 22.5 Step 0.5

OutputString = x & " - " & LGreq((x \* 2) - 12)

Displayreq.Items.Add(OutputString)

Next

End Sub

Private Sub Form1\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Dim loopControl As Decimal

For loopControl = 6 To 22.5 Step 0.5

lstTimes.Items.Add(loopControl)

Next

End Sub

Private Sub savedayplan\_Click(sender As System.Object, e As System.EventArgs) Handles savedayplan.Click

Using writer As StreamWriter =

New StreamWriter(Saveas.Text & ".txt")

For x = 6 To 22.5 Step 0.5

writer.WriteLine(LGreq((x \* 2) - 12))

Next

End Using

End Sub

Sub LoadLGreq()

Dim x As Integer = 0

Using read As New StreamReader(Loadas.Text & ".txt")

Do Until read.EndOfStream

LGreq(x) = read.ReadLine

x = x + 1

Loop

End Using

End Sub

Private Sub loaddayplan\_Click(sender As System.Object, e As System.EventArgs) Handles loaddayplan.Click

Dim OutputString As String

LoadLGreq()

For x = 6 To 22.5 Step 0.5

OutputString = x & " - " & LGreq((x \* 2) - 12)

Displayreq.Items.Add(OutputString)

Next

End Sub

Private Sub createdayplan\_Click(sender As System.Object, e As System.EventArgs) Handles createdayplan.Click

Day.allocateshifts()

End Sub

Private Sub viewdayplan\_Click(sender As System.Object, e As System.EventArgs) Handles viewdayplan.Click

Day.Printdayplan()

End Sub

Private Sub CreateDB\_Click(sender As Object, e As EventArgs) Handles CreateDB.Click

connstr = New MySqlConnection

connstr.ConnectionString = "server=127.0.0.1;user=root;database=lifeguards;port=3306;password=;"

Dim com As MySqlCommand

Dim q As String

Dim reader As MySqlDataReader

Try

connstr.Open()

MsgBox("Connection Successful")

q = "CREATE DATABASE lifeguards"

com = New MySqlCommand(q, connstr)

q = "CREATE TABLE `lifeguards`.`Timetable` ( `ShiftID` INT(5) NOT NULL ,`Payrol Number` INT(6) NOT NULL ) ENGINE = InnoDB;"

com = New MySqlCommand(q, connstr)

q = "CREATE TABLE `lifeguards`.`Rota` ( `ShiftID` INT(5) NOT NULL , `Day` TEXTNOT NULL , `Date` DATE NOT NULL , `Shift Start` DECIMAL(3,1) NOT NULL , `Shift End` DECIMAL(3,1) NOT NULL ) ENGINE = InnoDB;"

com = New MySqlCommand(q, connstr)

q = "CREATE TABLE `lifeguards`.`Lifeguards` ( `Payrol Number` INT(6) NOT NULL ,`First Name` TEXT NOT NULL , `Surname` TEXT NOT NULL , `Age` INT(2) NOT NULL )ENGINE = InnoDB;"

com = New MySqlCommand(q, connstr)

MsgBox("Done")

connstr.Close()

Catch ex As MySqlException

MsgBox(ex.Message)

Finally

connstr.Dispose()

End Try

End Sub

Private Sub SQL\_Click(sender As Object, e As EventArgs) Handles Sql.Click

connstr = New MySqlConnection

connstr.ConnectionString = "server=127.0.0.1;user=root;database=lifeguards;port=3306;password=;"

Dim com As MySqlCommand

Dim reader As MySqlDataReader

Try

Dim q As String = ""

connstr.Open()

MsgBox("Connection Successful")

If Action.Text = "INSERT" Then

q = "INSERT INTO lifeguards." & Table.Text & " (" & one.Text & ")" & " values(" & one1.Text & ")"

ElseIf Action.Text = "UPDATE" Then

q = "UPDATE `" & Table.Text & "` SET " & one.Text & " WHERE " & one1.Text

ElseIf Action.Text = "DELETE" Then

q = "DELETE FROM `" & Table.Text & "` WHERE " & one.Text & " = " & one1.Text

End If

com = New MySqlCommand(q, connstr)

reader = com.ExecuteReader

MsgBox("Done")

connstr.Close()

Catch ex As MySqlException

MsgBox(ex.Message)

Finally

connstr.Dispose()

End Try

End Sub

Private Sub loadshifts\_Click(sender As Object, e As EventArgs) Handles connstr = New MySqlConnection

connstr.ConnectionString = "server=127.0.0.1;user=root;database=lifeguards;port=3306;password=;"

Dim com As MySqlCommand

Dim q As String

Dim reader As MySqlDataReader

Try

connstr.Open()

MsgBox("Connection Successful")

q = "SELECT lifeguard.`First Name`, lifeguard.age, rota.`start time`, rota.`end time`" \_

& "FROM ((timetable INNER JOIN lifeguard ON timetable.`Payroll Number` = lifeguard.`Payroll Number`" \_

& "INNER JOIN rota ON timetable.`ShiftID` = rota.`ShiftID` WHERE rota.date = " & LoadLGs.text

com = New MySqlCommand(q, connstr)

While reader.Read()

Day.LG.Add(New TLifeguard(reader("`First Name`"),reader("age"),reader("`start time`"),reader("`end time`")

End While

MsgBox("Done")

connstr.Close()

Catch ex As MySqlException

MsgBox(ex.Message)

Finally

connstr.Dispose()

End Try

End Sub

End Class

Class TLifeguard ' a lifeguard on a particular day

Property Name As String ' .net auto implementaion of private field and public get and set

Property Start As Decimal

Property Ending As Decimal

Property Age As Integer

Property Task As New List(Of Integer)

Property ShiftWeight As Integer

Property needlunch As Boolean = False

Property TotalPoolSlots As Integer

Property TotalWorkingSlots As Integer

Property hourhalfmaxonpool As Boolean = False

Property LGWorking As Boolean

End Class

Class DayPlan

Enum Task As Integer

NotWorking = 0

Cleaning = 1

MainDeep = 2

MainShallow = 3

Teaching = 4

Lunch = 5

End Enum

Property LG As New List(Of TLifeguard)

Sub Printdayplan() ' outputs dayplan to a spreadsheet

Dim sw As New StreamWriter(Form1.PlanName.Text & ".csv")

Dim where As String = ""

sw.Write("TIME")

sw.Write(",")

For x = 0 To 7

sw.Write(LG(x).Name)

sw.Write(",")

Next

sw.WriteLine()

For Time = 6 To 22.5 Step 0.5

For j = 0 To 8

If j = 0 Then

sw.Write(Time)

Else

Select Case LG(j - 1).Task((Time \* 2) - 12)

Case 0

where = " "

Case 1

where = "C"

Case 2

where = "MP/D"

Case 3

where = "TP"

Case 4

where = "MP/S"

Case 5

where = "L"

End Select

sw.Write(where)

End If

sw.Write(",")

Next

sw.WriteLine()

Next

sw.Close()

Process.Start("C:\Users\William\Documents\Visual Studio 2013\Projects\tryagain\tryagain\bin\Debug\" & Form1.PlanName.Text & ".csv")

End Sub

Sub allocateshifts()

Dim TotalLGsworking As Integer

Dim valid As Boolean = False

Dim time As Decimal

Dim x As Integer

loadTestData()

Try

For time = 6 To 22.5 Step 0.5 ' for each half hour

For Each currentLifeguard In LG

If time >= currentLifeguard.Start And time < currentLifeguard.Ending Then ' if LG is currently working

currentLifeguard.LGWorking = True

currentLifeguard.Task((time \* 2) - 12) = Task.Cleaning

If currentLifeguard.Start = time Then

TotalLGsworking += 1

currentLifeguard.TotalWorkingSlots = 1

Else

currentLifeguard.TotalWorkingSlots += 1

End If

currentLifeguard.ShiftWeight = currentLifeguard.TotalPoolSlots / currentLifeguard.TotalWorkingSlots

'Try

'Catch ex As Exception

' currentLifeguard.ShiftWeight = 1

'End Try

ElseIf currentLifeguard.Ending = time Then

currentLifeguard.LGWorking = False

TotalLGsworking -= 1

Else

currentLifeguard.LGWorking = False

currentLifeguard.ShiftWeight = 999999 ' they won't get selected!

End If

Next

quicksortweight(LG, 0, 7) ' sorts the lifeguards

' pick the lifeguards

x = 0

If Form1.getLGreq((time \* 2) - 12) = 0 Then

For r = 0 To 7

If LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.Cleaning

End If

Next

ElseIf Form1.getLGreq((time \* 2) - 12) = 1 Then

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.MainDeep

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

ElseIf Form1.getLGreq((time \* 2) - 12) = 2 Then

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.MainDeep

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.Teaching

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

ElseIf Form1.getLGreq((time \* 2) - 12) = 3 Then

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.MainDeep

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.Teaching

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

Do

If LG(x).hourhalfmaxonpool = False And LG(x).LGWorking = True Then

LG(x).Task((time \* 2) - 12) = Task.MainShallow

LG(x).TotalPoolSlots += 1

valid = True

Else

LG(x).hourhalfmaxonpool = False

End If

x += 1

Loop Until valid = True

valid = False

End If

checkhourhalfmax(time)

Next

For l = 0 To 7

LGneedLunch(LG(l))

Next

Placelunch()

Catch ex As Exception

MsgBox("Not enough Lifeguards at - " & time)

End Try

End Sub

Sub checkhourhalfmax(time) ' looks at the past hour and half of a persons day tio see if they have been on poolside for 1.5 hours

For Each currentLifeguard In LG

If time >= currentLifeguard.Start And time < currentLifeguard.Ending Then

If currentLifeguard.Task((time \* 2) - 12) = Task.MainDeep Or currentLifeguard.Task((time \* 2) - 12) = Task.MainShallow Or currentLifeguard.Task((time \* 2) - 12) = Task.Teaching Then

If currentLifeguard.Task((time \* 2) - 13) = Task.MainDeep Or currentLifeguard.Task((time \* 2) - 13) = Task.MainShallow Or currentLifeguard.Task((time \* 2) - 13) = Task.Teaching Then

If currentLifeguard.Task((time \* 2) - 14) = Task.MainDeep Or currentLifeguard.Task((time \* 2) - 14) = Task.MainShallow Or currentLifeguard.Task((time \* 2) - 14) = Task.Teaching Then

currentLifeguard.hourhalfmaxonpool = True

End If

End If

End If

Else

currentLifeguard.hourhalfmaxonpool = False

End If

Next

End Sub

Public Sub quicksortweight(ByRef LGS As List(Of TLifeguard), LO As Integer, HI As Integer)

Dim x As Integer

Dim y As Integer

Dim mid As TLifeguard

Dim temp As TLifeguard

x = LO

y = HI

mid = LGS(Int((x + y) / 2))

Do While x <= y

Do While LGS(x).ShiftWeight < mid.ShiftWeight

x = x + 1

Loop

Do While mid.ShiftWeight < LGS(y).ShiftWeight

y = y - 1

Loop

If x <= y Then

temp = LGS(x)

LGS(x) = LGS(y)

LGS(y) = temp

x = x + 1

y = y - 1

End If

Loop

If LO < y Then Call quicksortweight(LGS, LO, y)

If HI > x Then Call quicksortweight(LGS, x, HI)

End Sub

Sub loadTestData()

For i = 0 To 7

For y = 0 To 33

LG(i).Task.Add(New Integer)

LG(i).Task(y) = Task.NotWorking

Next

Next

End Sub

Sub Placelunch() 'places someone on a lunch break as close to halfway through thier shift

Dim cleans As Integer

Dim lunchat As Integer

Dim a As Integer

For x = 0 To 7

If LG(x).needlunch = True Then

cleans = 0

For y = 0 To 33

If LG(x).Task(y) = 1 Then

cleans = cleans + 1

End If

Next

lunchat = CInt(cleans / 2)

cleans = 0

Do

If LG(x).Task(a) = 1 Then cleans = cleans + 1

a = a + 1

Loop Until cleans = lunchat

LG(x).Task(a) = Task.Lunch

End If

a = 0

Next

End Sub

Function LGneedLunch(thisLG As TLifeguard) As Boolean 'checks to see if someone need a lunch dependent on age and shift length

thisLG.needlunch = False

If thisLG.Age < 18 Then

If (thisLG.Ending - thisLG.Start) >= 5 Then

thisLG.needlunch = True

End If

Else

If (thisLG.Ending - thisLG.Start) >= 8 Then

thisLG.needlunch = True

End If

End If

Return thisLG.needlunch

End Function

End Class