# 1. Design

## 1.1 Database

Using the specification provided for the NHS Shift Management system, the key aspects were extracted to assist in the design on an appropriate system.

The system needs to store multiple users, in which each user belongs to a specific level. The nature of the system is to manage users shifts; as such a table is required to store which shifts each staff member is working.

With this in mind, a database structure was developed which fulfils all of these storage requirements, using 3 separate tables. Figure 1 shows a visual representation of the database structure.

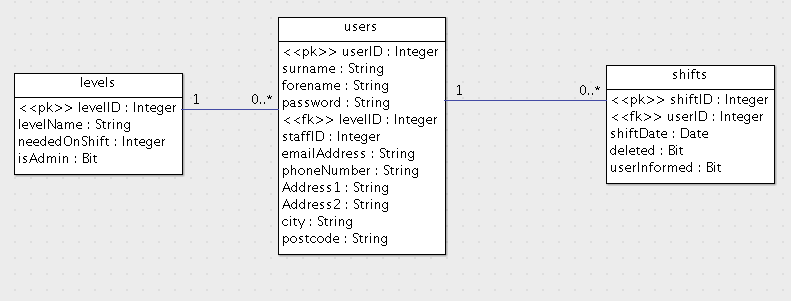


Figure 1 – UML database design

The users table (table 1) contains all of the user information, including the personal information, in one central location. There is a NHS staffID column that connects the staff member to their main NHS account. The levelID links the staff member to a specific staffing level.

**Table 1 – Users Table Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Description | Column Name | Data Type | Length | Nullable | Unique | Key |
| The unique, auto incrementing userID | userID | int |  | False | True | Primary |
| Staff members surname | surname | varchar | 200 | False | False |  |
| Staff members forename | forename | varchar | 200 | False | False |  |
| Encrypted Password | password |  |  |  |  |  |
| Staff levelID | levelID | int |  | False | False | Foreign (levels) |
| Staff NHS ID | staffID | int |  | False | True |  |
| Personal email address | emailAddress | varchar | 100 | True | False |  |
| Personal phone number | phoneNumber | varchar | 14 | True | False |  |
| Personal address | Address1 | varchar | 100 | True | False |  |
| Personal address | Address2 | varchar | 100 | True | False |  |
| Personal address | City | varchar | 100 | True | False |  |
| Personal address | postcode | varchar | 9 | True | False |  |

The levels table (table 2) is structured in a method that allows for future development of the system, with little to no modification of the existing system needed. One such development would be allowing for new staffing levels to be created (such as junior nurse).

The levels table links to the users table on the levelID.

The ‘isAdmin’ column determins if the staff level should be granted admin permission within the system, this allows for the possibility of senior staff being able to manage shifts.

The ‘neededOnShift’ column represents the minimum amount of staff needed to work a specific shift, modifying this in the DB reflects on the management tool but imposing the relevant restrictions.

**Table 2 – Levels Table Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Description | Column Name | Data Type | Length | Nullable | Unique | Key |
| The unique, auto incrementing levelID | levelID | int |  | False | True | Primary |
| The level name | levelName | varchar | 100 | False | False |  |
| How many staff of this level are needed as a minimum for a shift | neededOnShift | int |  | False | False |  |
| Determine if the level is granted Admin privileges. | isAdmin | bit |  | False | False |  |

The shifts table (table 3) contains each and every shift, for each staff member. If a shift is deleted, then it is kept in the tables but is marked as deleted. This allows admin to review deleted shifts, and it also allows for the deleted shift to be communicated to the staff member.

If any shift is created / deleted by admin, the userInformed column will allow the system to find these shifts, and then communicate the relevant message to the user.

**Table 3 – Shifts Table Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Description | Column Name | Data type | length | Nullable | Unique | Key |
| The unique, auto incrementing shiftID | shiftID | bigint |  | False | True | Primary |
| The user which is working the shift | userID | Int |  | False | False | Foreign (users) |
| The date of the shift | shiftDate | Date |  | False | False |  |
| If the shift has been deleted | Deleted | Bit |  | False | False |  |
| If the user is aware of the shift (Admin created) | userInformed | Bit |  | False | False |  |

## 1.2 Calendar

The calendar aspect will be created using the fullcalendar.io library. This is a JQuery library released under the MIT license, meaning it can be used in almost any way so long as credit to the creator is left in tact.

There is one main benefits to using FullCalendar, over creating a new calendar system; There is no need to reinvent the wheel. Multiple aspects are modifiable enabling FullCalendar to fulfil all of the functionality set out in the specification. Through modifying different parameters in the <script> section of the page, all functionality can be achieved, from changing the background colour of the events to handling the process of when a user clicks on a specific day.

# 2. Implementation

# 3. Testing

Test on a mobile device

# 4. Evaluation

# Bibliography