Student Number

Planning a database for the emergency services

**Contents**

[Table of Figures 2](#_Toc96038047)

[Introduction 2](#_Toc96038048)

[Determining the necessary data 3](#_Toc96038049)

[Victims 3](#_Toc96038050)

[Suspects 3](#_Toc96038051)

[Incidents 3](#_Toc96038052)

[Departments 4](#_Toc96038053)

[Regions 4](#_Toc96038054)

[Staff 4](#_Toc96038055)

[Shift pattern 4](#_Toc96038056)

[Vehicles 4](#_Toc96038057)

[The relationship between data 5](#_Toc96038058)

[The uses of CRUD 6](#_Toc96038059)

[Required features 7](#_Toc96038060)

[Switchboard 7](#_Toc96038061)

[Login system 8](#_Toc96038062)

[Reports 9](#_Toc96038063)

[Queries 10](#_Toc96038064)

[Additional features 10](#_Toc96038065)

[Staff timetable 10](#_Toc96038066)

[Staff allocation 10](#_Toc96038067)

[Appending and adding staff members 10](#_Toc96038068)

[Updating a form with specific data 11](#_Toc96038069)

[Test plan 12](#_Toc96038070)

[Bibliography 17](#_Toc96038071)

# Table of Figures

[Figure 1 Relationships between data 5](#_Toc96038072)

[Figure 2 the uses of CRUD inside the database 6](#_Toc96038073)

[Figure 3 Switchboard 7](#_Toc96038074)

[Figure 4 Login system 8](#_Toc96038075)

[Figure 5 Report of staff timetable 9](#_Toc96038076)

[Figure 6 Example of a complex query 10](#_Toc96038077)

[Figure 7 Example of a basic query 10](#_Toc96038078)

[Figure 8 Example of adding data to a table 11](#_Toc96038079)

# Introduction

The local emergency services team has recently requested for a list of developers, for a large-scale database to be created branching out for the entire South Yorkshire region for commercial uses. Inside this document is the entire process of the planning, including data that is required and the relationships between them with additional ideas that could aid the effectiveness and speed of the company.

# Determining the necessary data

Before building any database, planning is required to structure and decide what data is necessary. This assists with data redundancy and load times of the application, as this application and many others, if possible, need to remain as smooth and fluent as possible at all times to grant the most effective response time for staff deployment. A Report was conducted by (Nagy, 2015) in 2015 to assist in load times for an application the reasoning behind this application was as follows (Nagy, 2015)“If the system has more than one thousand users per day and it is still growing, load testing should be considered to be conducted by an expert or more thorough, by a group of experts”. This includes call handlers that would be on shift accessing the applications and updating data consistently. Depending on the circumstances it is possible that the amount of data being processed at the same time could cause downtime or delays on the software. This could cause issues with deployment if not planned for the situation if it was to occur.

After an in-depth discussion with the client, a list of data was created below is the list of data agreed upon.

## Victims

* First name
* Last name
* Contact Number

## Suspects

* First name
* Last name
* Contact Number

## Incidents

* Type of incident / Description
* Priority level
* Date and time of the incident

## Departments

* List of each emergency service

## Regions

* The area depots are located

## Staff

* First name
* Last name
* Department the work under
* Staff group
* Shift patterns
* Is the staff member on shift

## Shift pattern

* Days Wednesday to Saturday
* Nights Wednesday to Saturday
* Days Sunday to Tuesday
* Nights Sunday to Tuesday

## Vehicles

* Type of vehicles
* Registration number
* Which staff use this vehicle
* Is the vehicle at a job
* What sector is the vehicle for

# The relationship between data

A picture containing text, indoor, computer, screenshot

Description automatically generatedAfter the list of data was created into tables with the allocated data fields, the next step is linking the data together. The process this is achieved is by linking the relationships between the data this is displayed on (Figure 1). After creating an entity-relationship diagram for each existing table the primary keys are used to determine Id numbers.

Figure 1 Relationships between data

# The uses of CRUD

Graphical user interface, text, application, email

Description automatically generatedNow that the data is linked together, it is now possible to create functioning forms that can add new staff members, victims, suspects, and possibly new vehicles to the database. An example of this is shown in (Figure 2). The data that is inputted creates a new victim to the database and updates information such as a victim’s contact number which could be incorrect. After a set time, data should be removed and stored elsewhere for multiple reasons.

Figure 2 the uses of CRUD inside the database

The reason behind removing old data from the system is to remove no longer relevant information such as a resolved incident between a victim and suspect or a staff member leaving the workforce. Although this information is removed from the base it is still archived for necessary precautions. Normally this is stored online or at a data centre. However, (Arora, 2015) has stated that “the data movement is constricted due to network bandwidth and latency issues” therefore moving the data may need to be planned ahead of time-based. Upon the least common time of incident fillings, reducing the risk of overloading the system causing halts inputting data. Once this is achieved encrypting data could be viable to strengthen the security of the data from breaches, preventing hackers from discovering personal data such as names and contact details. This is supported by (Bongki Moon, 2002) that suggests using “SQL queries over encrypted data.”

# Required features

## Switchboard

The switchboard provides a simple, quick, and effective user interface that call handlers can navigate with ease. For this instance shown inside (Figure3 ) the call handler can select existing reports seeing which incidents have happened in which location. This helps the handler with navigating the entire database instead of searching for a specific list of data.

Graphical user interface

Description automatically generated

Figure 3 Switchboard

## Login system

Graphical user interface

Description automatically generatedBut a security measure has been put into place to prevent others who should not have access to the database (Figure 4). In the format of a login system when selecting the switchboard, it is required to log in before accessing data.

Figure Login system

## Reports

Multiple reports have been created throughout the build to display related data per each report. As stated above having the incidents be displayed as their own allocated locations helps discover which location is more common for what type of incident and how frequent incidents are. Another report displays the staff timetable stating which staff are on shift that day (Figure 5). This can be reviewed at the start of each shift assisting in working out how many vehicles are available at the start of a shift as well as how many vehicles are available during the shift giving a constant update providing constant time for adjustments.

Table

Description automatically generated

Figure 5 Report of staff timetable

## Queries

The way most of the reports function are by queries and complex queries an example of this is shown in (Figure 6) this complex query looks for which vehicles in the Doncaster area are not on a current job this is extremely helpful and efficient for the call handlers granting less time for searching for a vehicle to send out.

Graphical user interface, text, application, table

Description automatically generated

Figure 6 Example of a complex query

One of the basic queries created is the timetable query shown in (Figure7) the reasoning behind this query is so staff can be provided with their current timetable with ease this could be printed out and placed on the notice board at each depot.

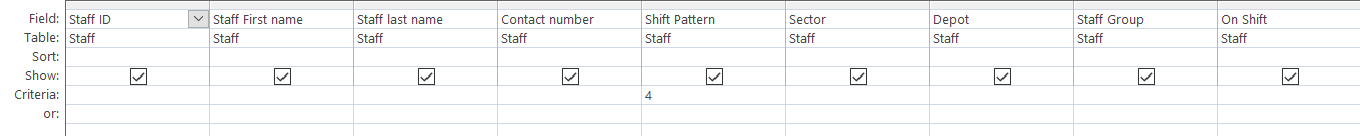


Figure 7 Example of a basic query

# Additional features

## Staff timetable

The ethical approach to this timetable was that each staff member works the same number of hours per week as staff need on call 24hours a day.

## Staff allocation

One of the features added to the system was the staff grouping which was listed in the NATO Phonetic Alphabet to prevent confusion this grouping system helps with viewing which staff are at work, which staff are on a job and what vehicle the staff member is using.

## Appending and adding staff members

This feature was suggested to update the database of staff members in case a member had left the company or requested to rotate to a new timetable or staff group, so a from was created to append the data.

## Updating a form with specific data

One of the queries created was to search for all incidents that happened in the Wath Upon Dearne region (Figure8). Once this query is run the data found is updated and added to a table that then could be viewed as a report format for printing this query could be added to each region or other existing forums.

Table

Description automatically generated

Figure 8 Example of adding data to a table

# Test plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Description | Data used | Expected result | Actual result | Pass or failed |
| 1 | Adding a victim to the incident list using the add victims query | Button click with an expectation of inputting data | When clicking the query, a message box appears letting the user input data such as names | When the query was clicked no box appeared but added a user named “Andy Frank” | **Failed** |
| 2 | Clicking the victims in the Wath query to update the existing table for printing. | Button clicks. | The table to be updated | Table was updated | **Passed** |
| 3 | Clicking depot not in job location Doncaster | Button clicks. | Display all vehicles that are not on a job in Doncaster | Displays correct data | **Passed** |
| 4 | Clicking depot in job location Doncaster | Button clicks. | Display all vehicles that are on a job in Doncaster | Displays correct data | **Passed** |
| 5 | Clicking depot not in job location Rotherham | Button clicks. | Display all vehicles that are not on a job in Rotherham | Displays correct data | **Passed** |
| 6 | Clicking depot in job location Rotherham | Button clicks. | Display all vehicles that are on a job in Rotherham | Displays correct data | **Passed** |
| 7 | Clicking depot not in job location Sheffield | Button clicks. | Display all vehicles that are not on a job in Sheffield | Displays correct data | **Passed** |
| 8 | Clicking depot in job location Sheffield | Button clicks. | Display all vehicles that are on a job in Sheffield | Displays correct data | **Passed** |
| 9 | Clicking depot not in job location Wakefield | Button clicks. | Display all vehicles that are not on a job in Wakefield | Displays correct data | **Passed** |
| 10 | Clicking depot in job location Wakefield | Button clicks. | Display all vehicles that are on a job in Wakefield | Displays correct data | **Passed** |
| 11 | Clicking depot not in job location Wath | Button clicks. | Display all vehicles that are not on a job in Wath | Displays correct data | **Passed** |
| 12 | Clicking depot in job location Wath | Button clicks. | Display all vehicles that are on a job in Wath | Displays correct data | **Passed** |
| 13 | Clicking the region query | Button clicks. | To display each total amount of vehicles onto a table | Displays 0 on each total amount | **Failed** |
| 14 | Clicking which staff members work Sunday to Tuesday 18:00 to 06:00 | Button clicks. | To display all staff on this timetable | Displays correct data | **Passed** |
| 15 | Clicking which staff members work Sunday to Tuesday 06:00 to 18:00 | Button clicks. | To display all staff on this timetable | Displays correct data | **Passed** |
| 16 | Clicking which staff members work Wednesday to Saturday 18:00 to 06:00 | Button clicks. | To display all staff on this timetable | Displays correct data | **Passed** |
| 17 | Clicking which staff members work Wednesday to Saturday 06:00 to 18:00 | Button clicks. | To display all staff on this timetable | Displays correct data | **Passed** |
| 18 | Changing data to existing staff members | Button clicks. | Form loads giving options to change data | Data is changed | **Passed** |
| 19 | Clicking adding new staff member form and adding a new staff member | Button clicks.  With an expectation of input data that will be added to the existing table | Form loads | New data is added | **Passed** |
| 20 | Clicking the incident report form | Button clicks. | Form loads  Giving options for first name, last name incident,  Date of the incident and contact number | Inputs new data | **Passed** |
| 21 | Clicking the login page and trying to log in | Button clicks. And inputting the required data | Login page to load and when inputting the correct data, the user logs in and gains access | An error message appears stating,  The object doesn’t contain the automation object ‘pass’ | **Failed** |
| 22 | Clicking the switchboard staff groups and seeing if the login feature pops up | Button clicks. | To display a warning message to tell the user to log in | The message does not show up and displays the data that should be hidden | **Failed** |

# Bibliography

Arora, R. (2015). *Data Management: State-of-the-Practice at Open-Science Data Centers.* New York, NY: Springer. doi:https://doi.org/10.1007/978-1-4939-2092-1\_37 / Online ISBN 978-1-4939-2092-1

Bongki Moon, D. D. (2002). *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data.* New York: Association for Computing Machinery. doi:https://doi.org/10.1145/564691.564717

Nagy, A. (2015). *Sensitivity Analysis and Patterns Implementation on Load Testing Software Systems.* Babeş-Bolyai University, Faculty of Economics and Business Administration, Computer Science. Romania: Journal of Software. DOI:DOI: 10.17706/jsw.10.12.1402-1411