Table of Contents

[Project Problem definition: 1](#_Toc117178670)

[Analysis: 2](#_Toc117178671)

[1. Background Research: 2](#_Toc117178672)

[1.1 Observation of Existing System: 2](#_Toc117178673)

[1.2 Problem Identification: 2](#_Toc117178674)

[2. Prospective End User Identification: 3](#_Toc117178675)

[2.1 Prospective User Interviews: 3](#_Toc117178676)

[2.2 Prospective User Acceptable Limitations: 6](#_Toc117178677)

[3. Algorithm Choice: 6](#_Toc117178678)

[3.1 Club Suggestion Algorithm: 6](#_Toc117178679)

[3.2 Club Cost Algorithm: 6](#_Toc117178680)

[4. Data collection: 6](#_Toc117178681)

[5. Proposed Feature List: 6](#_Toc117178682)

[6. Proposed System Objectives: 6](#_Toc117178683)

[7. Diagrams: 7](#_Toc117178684)

[7.1 Top level E-R Diagram: 7](#_Toc117178685)

[7.2 Hierarchy Chart: 7](#_Toc117178686)

[7.3 Top level DFD: 7](#_Toc117178687)

[STUFF TO DO: 8](#_Toc117178688)

# Project Problem definition:

Sutton Grammar School currently does not have a system in place for scheduling and choosing clubs. This program aims to rectify this by allowing students to enter their current clubs they are attending (if there are any) and then matching students to other clubs based on available timing and considering the subjects and clubs the student attends. This will be achieved using the Hungarian algorithm.

Clubs are sorted into different categories such as sports, academia, CCF, DofE, etc. There is also a category of clubs that can be chosen called “homework and detention” which students can choose or be forced to participate in if they have violated the behaviour matrix. Clubs contain data on club members, club presidents, teachers who run the club, the department the club runs in etc; each club also contains its own club description which have multiple attributes such as age/year range, description of what occurs, skills developed in the club, who it is run by etc.

The timetable of a student starts out empty, with 14 separate days (Mon A to Fri B) including all periods (in case of clubs taking place during free lessons/supervised study), lunch and afterschool. Students can then fill in any activities they are currently doing (if the club occurs multiple times, a specific time slot is specified). Based on the current students A-level subject choices and clubs currently attended (if applicable), students will be suggested clubs to attend, and given options as to what to do if suggestions conflict with each other or current clubs. Students cannot have clubs during scheduled lessons.

The timetable includes reports, which are written to/read from a text file. These reports include fields such as attendance, reasons for no attendance (ill, assembly, meeting etc) and general notes as to what happened in the club. Club presidents (who have their own separate login details) will have the ability to approve reasons for lack of attendance, as well as make an openly visible document that shows what was achieved during the club.

# Analysis:

## 1. Background Research:

### 1.1 Observation of Existing System:

The current club timetabling system is an online system in which a QR code of an authorised teacher can be scanned to sign a student into the club. The QR code contains personal student data, such as the student's first name, last name, and form. This data is appended onto the website URL, allowing for the student’s data stored in the QR code to be used on the system and sign a student in.

To have the ability to sign a student into a club, a user must scan the system administrator’s QR code. After filling out, some personal details, a cookie is stored in the device, which is checked against the set of registered cookies stored in a database to verify the user account. Then, only this subset of users with their unique cookie can have their QR code scanned by a club attendee, proving club attendance.

### 1.2 Problem Identification:

#### 1.2.1 Complications of Existing System:

The existing system, while functional, is inherently flawed in terms of data access.

1. Only an account manually given permission by Mr James (by scanning his “master QR code”) can allow students to sign into clubs
2. Data can currently only be accessed by Mr James, and is stored in a private database
   1. Club leaders cannot easily add, change, or remove attendee data
   2. Club leaders do not have access to a list of club attendees, and therefore cannot easily see who may have missed a club
3. The time periods given for club activities only includes mornings, lunchtime, and after-school
4. Each attendee must scan a club QR code, which could be time consuming and dependent on the QR code being in an easy to scan place
5. Club leaders or club attendees may both forget to sign into a club
6. The system lacks a form of club suggestion
7. The system lacks a form of club reports

Furthermore, while the security of the system itself is of merit, the attendee’s QR data is very easy to access. When scanning a QR code, the attendee data is appended to the end of the URL in plaintext, meaning that malicious individuals could scan the QR code for personal information on an individual.

#### 1.2.2 Benefits of Proposed System

The proposed system will instead rely on a program, in which all club leaders will be able to input data such as attendance, access and change attendee data. The program will also include a student-club matching algorithm, as well as the ability to add notes, notices, and reports will also be added. This system will also be linked to Mr James’ system, omitting the QR codes for a direct link to the website on which the existing system is running on.

#### 1.2.3 Proposed System Scope:

The result of my research into the problem has led me to the 2 underlying concepts behind the program’s scope. This will enable me to focus my interview initiatives as well as better regulate my system objectives.

##### 1.2.3.1 Student-Club Matching Algorithm

One rationale of the program is to match each user to an appropriate club, which considers both the subjects taken by the student, and – if applicable – any currently attended clubs.

##### 1.2.3.2 User Reports

The ability for users to add reports presents a versatile system, which can easily act as notices, notes, and notifications. The “Note Security Level” (NSL) to user reports are as follows:

|  |  |
| --- | --- |
| NSL | Who can access it? |
| Private | The user who made the report |
| Club | All users in the club that the report is related to |
| Department | All users in the department the report is related to |
| Shared (e.g. – with Department Head) | The user who made the report and the other shared user (e.g. – the club’s respective department head) |
| Public | All users |

|  |  |
| --- | --- |
| NSL | How is it encrypted? |
| Private | Uses the **private key** of the user who made the report |
| Club | Uses the **private key** of the club |
| Department | Uses the **private key** of the department |
| Shared (e.g. – with Department Head) | Asymmetric key encryption |
| Public | No encryption |

## 2. Prospective End User Identification:

### 2.1 Prospective User Interviews:

The main target demographics of end users for my application will be:

* Students attending clubs
* Club leaders
* Teachers associated to clubs

Since all 3 demographics have separate use cases, I prioritised speaking to one person from each to see what functionality they would value in a club management software. Furthermore, I also interviewed a schoolteacher, Mr James, who maintains the existing system.

#### 2.1.1 Interview with Mr James:

Given the already implemented features in the existing system, I prioritised to ascertain the workings of the existing system, as well as the data that could be available to me.

Attendees: RK = Rayan Khan (Interviewer), OJ = Mr Owen James (interviewee)

Date: Thursday, October 19, 2022

OJ: Welcome Rayan. I believe you wanted to interview me regarding the current club system at school

RK: Yes, thank you very much. First, I would like to ask you to give me a simple explanation of the current system.

OJ: Sure! Teachers are registered as club leaders by scanning my QR code. This leads them to a sign-up page where they input some information like their name. This stores a cookie on their device which is used to verify their identity. Then club leaders can scan the club attendee’s QR codes to sign pupils into a club.

RK: That sounds like a good system. But could I ask why you chose QR codes?

OJ: Well, the system needed to be flexible, and without a formal register – which will be harder to use since clubs are optional – I wanted the most frictionless answer possible. So I landed on QR codes. They are easy to store data in and, after COVID especially, everyone knows the basics of how to use a QR code.

RK: Do QR codes not make club leader registration and club attendee sign-in a long and gruelling process though?

OJ:

RK: On another note, what if you or the student forgets to sign in the club attendees?

OJ:

RK: And what would happen if the club leader removed the cookie from their device?

OJ:

RK: And finally, I would like to ask a couple questions about data access and the databases you are using.

OJ: Sure.

RK: First, can the data only be handled by you?

OJ:

RK: Next, how much data is available as test data to a student like me?

OJ:

RK: Finally, I would like to ask you to explain the structure of your databases?

* Explanation of existing system

Register teachers as club leaders

Pupils have qr coders

Club leaders scan qr codes pupil attends club

* Why did you choose existing system
  + Why QR codes?

System needs to be flexible

Cant do formal register

* Database structure
  + Is it normalised?

5 databases

1. Qr code database
   1. Name
   2. House
   3. Initials
   4. Timestamp
2. Timestamp referenced against club database (scanned against OJ Monday lunch is electronics sign)
3. List of device registered for double checking (cant copy cookie, have to see OJ)
4. Login database
5. Database staff codes (OJ)

* How much data is available

Historic data

* Data being handled by only 1 person

Data only visible to Mr James

Once system finished, accounts can be allowed to login

* Teacher registration is gruelling and long?

Scan jame’s qr code, initials in and cookie stored

Cookie reset/delete

* fail to scan students in

house competition for most extracirricular attendance

motivator for kids to remind teacher to sign kids in

scanning in cant be compulsory so QR codes made to be as frictionless as possible

#### 2.1.2 Ms Sharon Corkery:

#### 2.1.3 Mohammed Keeka (Year 13):

### 2.2 Prospective User Acceptable Limitations:

6th form students

## 3. Algorithm Choice:

### 3.1 Defining the Correct Algorithm Subset to Use for Club Assignment:

Algorithm classifications are many, and so it is important to identify which subset of algorithms it will be best to choose an algorithm for. The foremost algorithm I thought of was the “brute force” algorithm. Here, every combination (*or in context, every club suggestion/timetable*) is tested to see which is the most superior. While the main benefit of this paradigm is that the definitive best solution is achieved, the extreme time complexity (time taken for the algorithm to run) has made me decide that a faster algorithm, while it may be less definite in solution, is a better solution.

However, my research into brute forcing enabled to me to discern that the problem is in fact an **optimisation** problem – that is, the problem of finding the best solution from the set of all possible solutions. Specifically, since the problem is required to solve the assignment of a combination of clubs, each of which occur at a specific time (and that student may have some bias to), the collection of **combinatorial optimisation** problems is more accurate. From here, the problem can be modelled as a particular combinatorial optimization problem – the assignment problem. In simple terms, the problem is as follows:

The assignment problem has several agents and several tasks. Any agent can be assigned to any task, through which a cost is applied. The cost varies between agent-task assignments (between the combinations of agents and tasks). It is required to perform the maximum number of tasks possible by assigning each task to one agent in such a way that the total cost is minimised.

### 3.2 Club Suggestion Algorithm:

In terms of the assignment problem, we can model *agents* as *potential club attendees* and *tasks* as *clubs to join*. Each *task* incurs a cost, which – in this case – is modelled as the bias a student has towards clubs based on their current subjects and any currently attended clubs. This cost algorithm is further detailed in section 3.2, the “Club Cost Algorithm”.

#### 3.1.1.1 Branch-And-Bound Algorithms

While initially researching a complementary algorithm to solve the assignment problem, I stumbled upon branch-and-bound algorithms. The goal of a branch-and-bound algorithm is to find a value that maximises or minimises a real-valued “objective function” (in the case of our model, minimising the cost), among a set of values that are in the set of solutions. It achieves this by recursively reducing the search space (by creating subproblems of the overall optimisation problem) and minimising the objective function for the smaller space. This part of the process forms the *“branching”* in the branch-and-bound algorithm. However, just branching would be the equivalent of brute forcing solutions, which I have already decided not to do. To improve on this, the algorithm keeps a track on the *“bounds”* of the minimum cost value it is trying to find and uses these bounds to shrink the search space further, avoiding sub-trees of possibilities that do not contain an optimal solution.

*Essentially, Branch-and-bound algorithms are very similar to brute forcing a solution, with the added benefit of “intelligently” selecting the possibility with the least cost*

|  |  |
| --- | --- |
| Pros | Cons |
| * Relatively simple to code * Does not explore all combinations * Does find the optimal solution | * Very time consuming * Not scalable to larger problems without taking an unreasonably long time * Very difficult to parallelise |

#### 3.1.1.2 The Hungarian Algorithm/Munkres Assignment Algorithm

Continuing my research, I also found the Hungarian algorithm. The Hungarian algorithm is a solution to the assignment problem that solves it in polynomial time – O(n3) – compared to the typically exponential time complexity of branch-and-bound algorithms – O(2n). In simple terms, it is much, much, faster. It can easily be modelled as a matrix, and several calculations can be run to the matrix to materialise the optimal solution.

|  |  |
| --- | --- |
| Pros | Cons |
| * Can consider multiple resources as “cost” * Considerably fast (especially compared to branch-and-bound algorithms) | * “Cost” must be calculated separately * Slightly difficult to understand and code |

#### 3.1.1.3 Genetic Algorithms

Genetic algorithms are a subset of algorithms that generate solutions to optimisations and search algorithms by relying on biologically inspired factors, such as natural selection. It is also defined as a “metaheuristic” algorithm – that is an algorithm that provides an iterative approach which combines concepts to find, generate, or select a heuristic algorithm/solution. A heuristic algorithm is one that provides a sufficiently good solution within a reasonable time frame.

|  |  |
| --- | --- |
| Pros | Cons |
| * It is the best of all worlds, due to its metaheuristic nature * It has the possibility to be the quickest | * Extremely difficult to code * Does not provide a definite “best” solution |

#### 3.1.1.4 Chosen Club Suggestion Algorithm

Between the 4 choices – brute force, branch-and-bound algorithms, the Hungarian algorithm, and genetic algorithms – I have decided to choose the Hungarian algorithm. Brute forcing and branch-and-bound are too slow, whereas genetic algorithms are too complex. Furthermore, genetic algorithms only find a sufficient solution, whereas a student can be expected to want the best solution possible. Furthermore, the modelling of the problem lends itself well to the Hungarian algorithm, especially when considering the cost.

### 3.3 Club Cost Algorithm:

The “cost” (which is the input into the Hungarian) represents how close a club is to a student’s current subject choices and/or clubs currently attended. This link between choices can be represented as a graph, and the “cost” could be conceived as the minimum cost needed to traverse the graph, from the club to the subject.

Furthermore, to retain the authenticity of all subjects being related, even indirectly, I have created a categorisation of subjects called “meta-subjects”. Meta-subjects are groups of subjects that are related to each other; the list of them includes: sciences, humanities, languages, arts.

#### 3.3.1 Choosing an Algorithm to Calculate the Cost

#### 3.3.2 Student Willingness

The final input into the “cost” is the willingness of a student to attend a club. This acts as a score out of 10 (with 1 being the most willing and 10 being the least). The answer (*x*) is represented as a fraction . This becomes a multiplier to the previously calculated cost – for example, an answer of *1* will give a fraction of , meaning the previous cost is multiplied by 0.1. On the other hand, an answer of *10* will give a fraction of , meaning the previous cost is multiplied by 1 and stays the exact same.

## 4. Data collection:

## 5. Proposed Feature List:

## 6. Proposed System Objectives:

1. Student-club matching based on available timings
   1. An algorithm is used to assign a student’s prospective clubs when considering the “student cost”
      1. The “student cost” matrix (set of values) is calculated by considering:
         1. Student availability (lunch lessons, clubs that can run during period 6 and 7, etc)
         2. Student willingness – a score out of 10 that acts as a decimal multiplier to the overall student cost
         3. The student’s current subject choices and (*if applicable*) any clubs that the student is currently attending
   2. Another algorithm is used to find the most related clubs, subjects, and meta-subjects (high level subject grouping classifications), which is used as an input the Hungarian’s student cost matrix. Different connections have different associated costs:
      1. Links between club and club has the weakest connection and therefore the highest cost
      2. Links between meta-subject and club suggest an indirect/generalised connection and therefore have a middling cost
      3. Links between subject and club are direct and therefore have the lowest cost
      4. Meta-subjects are generalised guidelines to minimise the number of connections (preventing every subject/club from connecting to every other subject)
   3. The ability for students to select one club over another if wanted
2. School lesson/free period timetabling
   1. Using the school’s timetable to deduce scheduled lesson time and supervised study periods
      1. Teacher accounts are given full administrative permissions to change all data
3. Club reports system
   1. Each account type can submit reports on the club
      1. Students accounts can submit notes and give feedback on the club
      2. Advanced students can do all the above as well as report on specific students, attendance and give out notices that can be seen by all the other students
      3. Teachers can do all the above as well as add larger scale department notices
      4. Notices can also be sent to the teachers assigned to clubs/the head of the department, either as
      5. Reports can be encrypted based on attributes such as a studentID, clubID, departmentID, etc. These IDs are unique per student/club/department, and so one can encrypt and decrypt a report with a corresponding ID in order to make a file visible to a student/club/department
4. Security
   1. Secure login system
      1. Student, Advanced student (6th form student) and teacher accounts
         1. Students allowed basic user access to the system
         2. Advanced students are given semi-administrative permissions to change club data, although teachers are notified

## 7. Diagrams:

### 7.1 Top level E-R Diagram:

### 7.2 Hierarchy Chart:

### 7.3 Top level DFD:

# STUFF TO DO:

* Analysis
  + ER diagram
  + DFD
  + Background research
    - Current system
      * Environment of current system
      * Flowchart of current system
      * Evidence of current system
    - Problem identification
      * Existing system vs current system
      * Tasks to be automated and benefits of automation
        + Objectives of proposed solution
      * Potential solutions and justification of chosen solution
    - Objectives for the proposed system
  + Interview with user
    - Ms Corkery
    - Greenpower
    - Ritvik
    - Yuvan
    - Every0ne
    - ETC
  + Prospective users and acceptable limitations
    - User identification
    - User requirements
    - User needs
    - Other limitations
* Design
  + Overall system design
    - System flowchart
      * Login
      * Main menu
      * Adding/editing/removing students, clubs, teachers etc
      * **Notification system (EMAIL/WINDOWS NOTIFICATION)**
  + DFD
  + Data dictionary and validation
  + Justification of included Q types
  + Stepwise refinement
  + Database normalisation & design
    - E-R diagram
    - Possible SQL Queries
  + Top down design
  + OOP class design
  + Definition of record structure
  + System security & integrity of data
    - Password protection
    - Encryption
  + File organisation
  + UI design
    - Visuals
    - Navigation
    - Output
  + Algorithm design
    - Base algo
    - Modifications to base algo
    - Searching
    - Validation and authentication
    - Importing/exporting to txt file
    - Repeated data entries
  + Testing **plan**
    - Input and output testing **design**
    - Navigation testing **design**
* Testing
  + Input and output testing
    - Normal data
    - Erroneous data
    - Data OOB
  + Trace tables
  + Navigation testing
  + TESTING SCREENSHOTS FOR EACH CATEGORY
  + Test each individual component of NEA
* Technical solution
  + Detailed algo design
  + Class overview
  + Subrouting and variable overview
  + Post implementation system overview
  + System flowchart
  + EXPLAIN HOW EACH FILE OF CODE WORKS, WITH ITS SUBROUTINES, CONNECTIONS etc LINKED ALL TOGETHER
* User guide to program use
* Appraisal
  + Comparison of project performance against objectives
  + Potential future developments
  + Client feedback
    - Ease of use
    - How does the system meet the objective
    - Ease of setup
    - Criticisms
    - Improvement
    - Evaluation of client
    - Implementation of user feedback if time allows
* Appendix
  + Normalisation

<https://docs.google.com/document/d/1bnPV4poTgU_rfHnOuppRkhYm00UPKckAb08-aUFBwCM/edit#heading=h.qjisk72r9mlm>

<https://docs.google.com/file/d/0B58fq8yIsMahNEtPVkNXVUdHd00/edit?resourcekey=0-CLMprHJsUDqcqVtnfeAkig>

<https://www.modernanalyst.com/Resources/Articles/tabid/115/ID/2117/Defining-Scope-with-Feature-Levels-and-Events-Scope-Part-2.aspx>

<https://docs.google.com/document/d/1_jFrnsnQfYy5B2jcw2Dur-8st7IIhKjL/edit#heading=h.gjdgxs>

* Clubs sorted into categories
  + - Sport
    - Academic
    - School productions (drama)
    - Homework club & detention club
    - CCF
    - (functionality for students helping teachers out in lessons?)
    - DofE
      * After a number has reached a specific number of strikes, the corresponding student is assigned a detention time (given the option of changing if not working) and day based on available times
      * Cannot coincide with club to ensure extracurriculars remain for students
        + This can be overridden if necessary
    - Etc
  + students are suggested to do a club of minimum X types
* list of all teachers running clubs, when it is running
  + clubs **can** occur during supervised study
* all clubs have multiple fields of descriptions
  + age/year range
  + basic description of what is generally done
  + skills developed in club (hard + soft)
  + run by who
  + corresponding department
* AUTO SUBMIT TO MR JAMES’ website if showed up
* Timetable has reports
  + Txt files which are written by students as notes
  + Shows attendance, signed off by club leader with code
  + If no attendance, a reason can be specified
    - Reasons can be categorised differently, eg:
      * Ill
      * Exeat
      * Assembly
      * Etc
  + Club leader can submit report of what was achieved during club if wanted
* Basic login which display details of:
  + Clubs and yourself (if member of club)
  + All club members (if head of club)
* Ability to add/remove/change clubs, details, teachers, associated departments, club category etc

<https://www.sqlite.org/see/doc/trunk/www/readme.wiki>

<https://github.com/Willena/sqlite-jdbc-crypt>

the assignment problem is a variation of the transportation problem with two characteristics

1) The cost matrix is a square matrix

2) The optimum solution for the problem would be such that there would be such that there would be only one assignment in a row or column of the cost matrix.

This method works on the principle of reducing the given cost matrix to a matrix of opportunity cost. Opportunity cost here shows the relative penalties associated with assigning resource to an activity as opposed to making the best or least assignment.