**Requirements Analysis**

We were required to build a database system that will allow the SMMC to perform their business operations quickly, efficiently and accurately. We have focused on the operations that we think are the most essential, and haven’t implemented some that we think are out of scope.

We have not tried to create payment processing, and only keep record of if a student has paid for their enrolment or not. If we had more time we would try and link our application with an external system which could handle enrolment payments and payroll.

We have also not implemented authentication pages and staff. In the future, we would want to have different staff members have access to specific pages, such as the Technicians only being able to access repairs and instruments. Currently one admin user performs all of the operations.

Our main focus in our application has been around keeping track of Students and Enrolments, and all of the other entities that relate to these. We have tried to make the user experience as smooth as possible, and minimize data entry and have the system do most of the work.

**Student**

* View generalised information on all students (name, dob, address, contact details)
* Create student with the necessary information (name, dob, address, contact info, guardian info)
* Create contact for student if over eighteen
* Create guardian and contact and link to student if student is under eighteen
* Display guardian or contact inputs fields live on student create based on date of birth entered
* View details of each student individually, along with all the instruments they are enrolled in
* Delete student and any subsequent instances of student in other entities

**Staff**

* View generalised information on all Staff (name, address, hours, contact details)
* Create staff and contact with the necessary information (name, dob, address, phone number, email, start date, hours working)
* Edit all information on staff (name, dob, address, phone number, email, start date, hours working)
* View details of each staff member individually
* Delete staff and any subsequent instances of staff in other entities

**Tutors**

* View generalised information on all Tutors (name, address, hours, contact details)
* Create tutor and contact with the necessary information (name, dob, address, phone number, email, start date, hours working)
* Edit all information on tutor (name, dob, address, phone number, email, start date, hours working)
* View details of each tutor individually,
* Show list of what instruments they can teach
* Show list of instruments currently teaching
* Delete tutor and any subsequent instances of tutor in other entities

**Tutor Type**

* View generalised information on all instruments all tutors can teach (tutor name, instrument, proficiency/grade)
* Create tutor type by selecting from lists of existing instances of tutors, instruments, and grades
* Edit all information on tutor type (tutor name, instrument, proficiency/grade)
* View details on each tutor type individually
* Delete tutor type

**Technician**

* View generalised information on all Technicians (name, address, hours, contact details)
* Create technician and contact with the necessary information (name, dob, address, phone number, email, start date, hours working)
* Edit all information on technician (name, dob, address, phone number, email, start date, hours working)
* View details of each technician individually
* Delete technician and any subsequent instances of technician in other entities

**Performance**

* View generalised information on all performances (date, time, venue)
* Create performance with the necessary information (date, time, venue)
* Add new venue
* Link all ensembles to newly created performances
* Edit performance information (date, time, venue)
* Add a piece to the performance
* Remove piece from performance
* View details of each performance individually, along with the ensembles that will be performing
* Delete performance and any subsequent instances of performance in other entities

**Ensemble**

* Generate ensembles by assigning every enrolled student to one of the three ensemble groups based on their highest enrolled grade

**Piece**

* View generalised information on all pieces (title, composer, last performed date)
* Create a new piece
* Delete a piece

**Payroll**

* Generate staff payment estimation.
* Pay senior and junior tutors using different hourly rates, calculate by using their working hours and checking their most prominent tutor role (senior or junior).
* View generalised payroll history of all payrolls (staff name, date, amount paid out and notes containing the role they are being paid for)
* View details of each individual payroll
* Delete instances of payroll

**Enrolment**

* Enrol student by selecting from lists of existing instances of students, instruments, and grades.
* On create, include a checkbox to specify if loan is needed and auto a loan instance on an instrument for current enrolment if selected. This is handle by the system and doesn’t need to be manually done.
* View generalised information on all enrolments (student name, instrument, date and grade)
* When a student is enrolled in an instrument, automatically add them to a suitable class if it exists. If it doesn’t exist, the system checks if three students have been enrolled in the same instrument and grade. If three people have been enrolled, then create a class with a suitable tutor, and create instances of MusicClass Enrolment to add students to the class. If there aren’t enough students to make a class, do nothing and the class will be created when more students are enrolled. This is all handled by the system and doesn’t need to be manually done.
* View details of each individual enrolment.
* Edit enrolment information (student, instrument and grade).
* Delete instances of enrolment.

**Classes**

* View generalised information for all classes (start date, end date, lesson times, instrument, grade, tutor)
* Edit class instance to assign tutor if no tutor has been assigned for the class instance, select from list of tutors if there are any available.
* Edit class instance to remove students from class.
* Add students to the class manually.
* View details of each individual class, along with a list of students enrolled in the class
* Delete instances of class
* Edit lesson time of class.

**Instrument**

* View generalised information existing instruments (name, student fee, open fee, hire fee and head tutor)
* Insert new instrument providing instrument name, student fee, open fee, hire fee and selecting a head tutor from a list of tutors
* Edit instrument information (instrument name, student fee, open fee, hire fee and tutor from a list of tutors)
* View details of each individual Instrument
* Delete instances of Instrument

**Local Musicians**

* View generalised information for all local musicians (name and contact details)
* Create new local musician with contact
* Edit local musician details
* View details of each individual local musician
* Add local musician to the community orchestra ensemble
* Delete instances of local musician

**Instrument Loans**

* View generalised information on all instrument loans (date, enrolment details, and instrument).
* Create new loan by selecting an enrolled student and an instrument from a list of available instruments in the inventory.
* Return an instrument.
* Send instrument for repair and assign a technician to the repair.

**Instrument Inventory**

* View generalised information of instrument inventory (received date, manufacturer, cost, and instrument).
* Edit instrument inventory information (received date, manufacturer, cost, instrument)
* Delete an instrument from Inventory
* Add an instrument to inventory (received date, manufacturer, cost, and instrument). This includes a quantity field that will create multiple instance of Instrument Inventory.

**Repairs**

* View generalised information of all instruments in repair (instrument name, technician , instrument)
* Edit repair to mark instrument as repaired/complete
* Issue replacement loan if available
* View list of all completed repairs.

**Sheet Music**

* View generalised information of all sheet music (title, composer, number of copies available for loan)
* Create a new instance of sheet music with the necessary information
* Edit sheet music information
* View details of each individual sheet music
* Delete instances of sheet music

**Sheet Music Loans**

* View generalised information of all sheet music loans (amount of copies loaned, sheet music details, tutor loaned by, date)
* Loan sheet music by selecting from a list of sheet music available, the tutor being loaned by and the quantity and decrement sheet music copies available for loan
* Return loaned copies of sheet music and increment sheet music copies available for loan