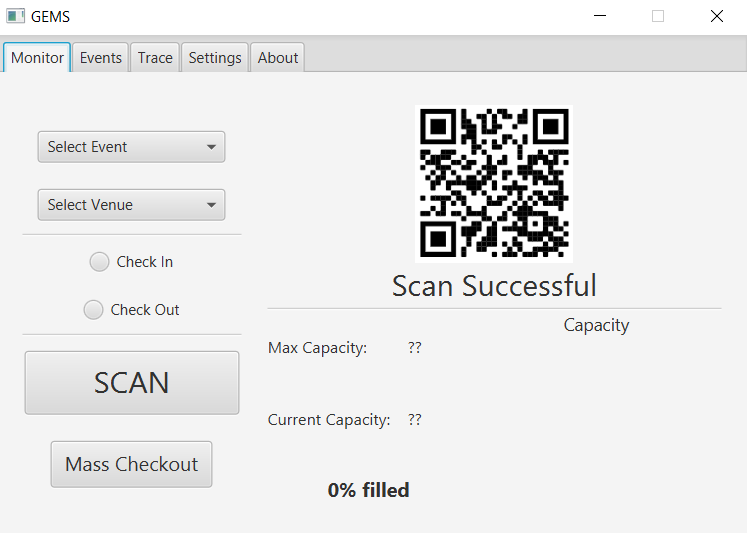
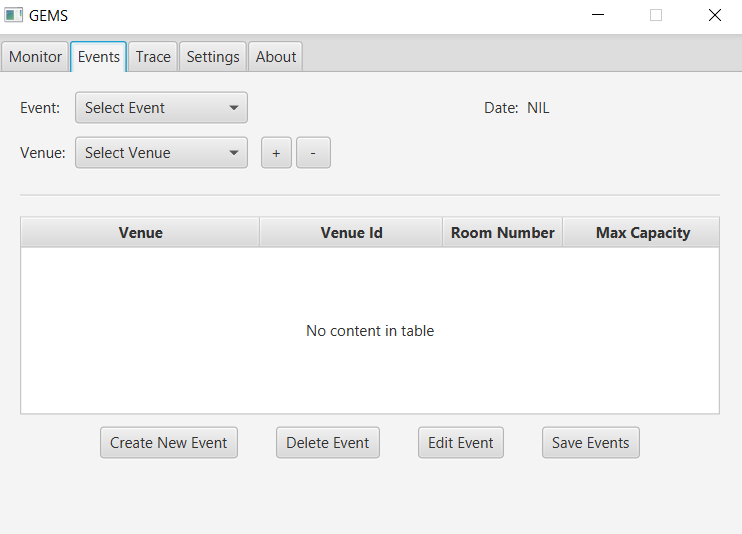
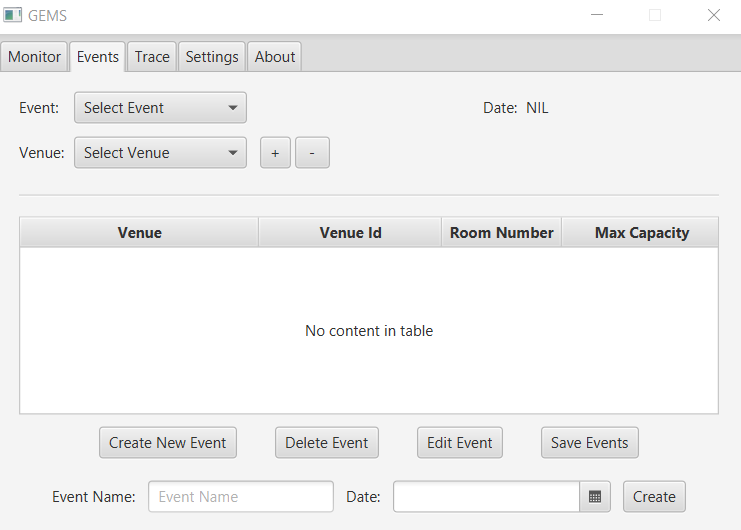
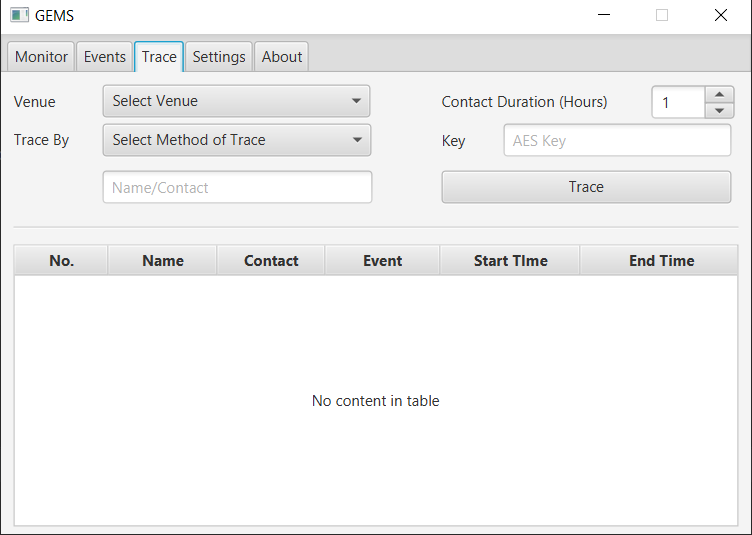
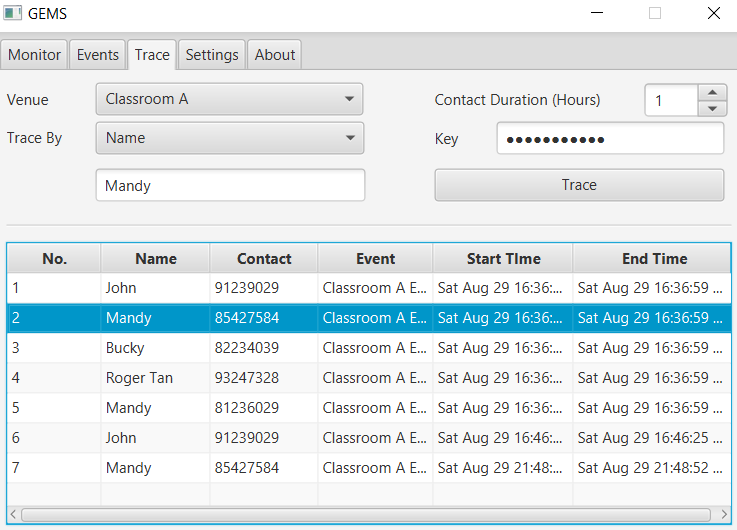
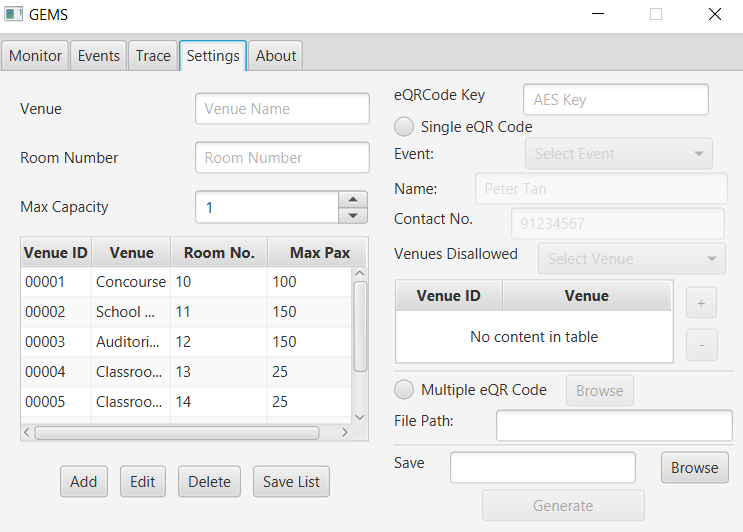
Edden Chew (7) M20306

We start discussing the key features from Venues, which is located in the Settings Tab. The Functionalities here are the creation, editing, and deletion of Venues. The user input which it requires are the Venue Name, Venue Room Number (restricted to integer and character input), and Maximum Capacity. The program allows for the saving of these venues into a csv file, which is loaded in when the program starts. This ensures that the venues still have the same VenueID. Note that no two Venue IDs will be the same, as the new VenueID will be loaded from a txt file. We also note that no two Venues may have the same name. Notably, a venue cannot be removed if it is already registered to an event.

In the Events Tab, Events may be created. To create an event, the “create event” button is clicked. Then, more options will be shown, allowing one to input the Event Name and Date of the Event. Upon pressing the “Create” Button, the Event will be created. From here, one may also select an event, and use the + and – buttons to add and remove venues from the event. To do so, one selects the venue from the combobox, and then selects the + or – button to add, or remove the venue (respectively). Furthermore, one may delete the event, or edit the event (one may change both the event name and date here)

Next, generation of QR Codes. Here, each QR Code represents a person. This is how the people ‘identify’ themselves to the program. In here, there are two ways to create QR Codes: mass creation and individual creation. In individual creation, a person’s name, contact number, disallowed venues, and event are collected. Then, the location where the image file will be saved is also collected, and from there, upon clicking the generate button, the eQRCode will be generated. We note that these eQRCodes are encoded with the AES Key stated at the top of the program, ensuring that the person’s details will not be exposed anyone who scans the QRCode. As of current, the only supported AES Code is “NUSHigh2020”, this also ensures that only people who know the AES code can add people.

Furthermore, eQRCodes may also be mass generated, which involves the loading of a csv file, with the format: “<eQRCodeID>, <name>, <contactNo>, <disallowedVenues>”. If the file is not in that format, then the file will be rejected, and no eQRCode will be generated, lest there be any corruption of the data.

In the monitor Tab, one may select an event and one of the venues that the event uses. Then, one may use the computer to “Scan” their QR code, by selecting an image. If the person is checking in, the program will check if the person is allowed to go to that venue or not; if so, the person will be checked in and logged into a database. If the event that the user is checking into is not on that day, user will receive a confirmation as to whether they want to check in to the event. For checking out of a person, the program checks if the person is logged into the database and is logging out for the correct Venue and Event. If so, the user is checked out. Finally, one may mass checkout all people from a venue, by merely selecting the venue and Event. Only if the user is participating in that event and venue will the person be logged out. Note that upon program closing, all people will be automatically checked out (Also all events and venues logged).

Finally, the trace tab. This tab allows for one to track the interactions between people. Given a person’s particulars, the venue which we want to restrict the search to, the correct AES Key (NUSHigh2020), and the duration which we want to track the person for, we can generate all people who may have interacted with that person in the venue. For example, assume that someone called Mandy may be infected, and we want to trace all the people who entered the venue from the time Mandy entered the venue, to 1 hour after that time. Given this, the program generates ALL the people who did so, including Mandy herself. This ensures that if there are more than one “Mandy”s on the database, both will be displayed, ensuring that no “Mandy” is hidden from the Database. This, unfortunately, makes the assumption that this may occur. To remove all people with the name “Mandy”, line 879 in SampleController may be uncommented. This line will remove all people with the same **name** as the people which the program is checking. My program also assumes that the user does not delete any events after they are held, as if this is done, nothing will pop up on the trace; the whole program as a whole then just becomes very buggy. Realistically, users will have no rationale to do this.

Note: the following 2 lines are in sampleStud.csv

240720VBAX00001,Mandy,81236029,00000 (line 2)

240720BSUD00001,Mandy,85427584,00000 (line 5)