- The whole project uses the façade as the design pattern, which is the class named ViewController, this class can instantiate business controllers and customer controllers, therefore the main class named AppointmentBookingSystem only need to instantiate two constructors: DatabaseManager and ViewController, which is convenient for any changes.
- All fxml pages were created in Scene Builder and have their own controllers, so all controller classes are parallel to each other, except the ViewController.
- The database contains 6 tables and all information are stored in it. More details can be found in "[SEPT] Database v1.png".
- Tables in the database use various values as primary keys. Such as business owner using owner_username and customerinfo using customer_username. So the program allows each unique business owner and customer, a business owner can add bookings for existing customers, a customer can choose a business for bookings. This is not always the case in reality, but the customerinfo doesn't contain any foreign keys related to their selected business owners, and this isn't specified in the assessments, so the program allows a customer to select businesses.
- OwnerCreateBookingController, CustomerCreateBookingController, CustomerViewAvailabilityController and OwnerAddEmployeeController have methods named eith CheckTime or CheckTimeSlot. To check availability for both business owner and customer, time in a day has been split into 48 slots for each 30 minutes. Time slots are defined by HashMap<Integer, booean>. The method sets all slots to false initially, first check if the employee works on the date, set working time slots to true, and check bookings for this employee, set existing booking slots to false. So the final available time list will be shown. One thing need to mention is that

there's also a method checking if a booking is within business hours. This is the case that when a business owner updates their business hours but forget to update employees' working time, even though the customer has chosen the time for an employee, but still cannot make a booking since the business doesn't run at that time. Same method has been used to check employees' updated working time are within business hours.

- There are four object classes: BusinessOwner, Booking, Employee and BookingAvailability. When the program reads data from the database, it will create relevant TableView in GUI, so these classes are used as holders for TableColumn, e.g. Booking <2017-1-2, 12:00, Andrew Dion, Men's Haircut, Ran Lu, 0123456789>. These objects are not created before selecting functions containing table views, and will be eliminated after this.