**Dental Appointment System Client-Server**

1. **Three Main Features**
2. **Client-server** – a client-server is set up for the dental appointment system which the user can book or edit their appointment using the browser. There are 9 handlers for the different routes to handle the different functions at the backend.
3. **Cookies –** cookie is used to create a session for the users when they logged in and will be deleted when they logged out.
4. **Templates –** templates are used to shown the output to the users on the browser whereas the logic are computed at the backend. The results computed at the backend will be passed to the templates as data to be shown to the users on the browser.
5. **Client-Server Description**

In the dental appointment system, there are total of 5 features, 3 features for patient and 2 features for admin usage which are elaborated in the “Dental Appointment System” word document on how each function works. However, in the client-server, the edit appointment function has also been added to the admin side which has the same function as the patient.

For the client-server, there are 3 users (“jpschew”, “ivan”, “ken”) and 1 admin (“admin”) initialized for testing purpose. The password for each of them can be found in server.go at initialize function.

There are 9 routing available for the client-server to handle.

1. **Home route (“/”)** – this is the home page for the dental appointment system where it asks the users to log in to their account or sign up for a new account.
2. **Signup route(“/signup”)** – this is the page for users who want to sign up for new account. User will need to key in a username, password, first and last name to sign up for an account.
3. **Login route(“/login”)** – this is the page for users to log in their account. A username and password need to be keyed in here.
4. **Make appointments route(“/makeAppt”)** – this is the page for users to make an appointment. Users will need to key in the date and time for their appointment. A list of available dentists will then be shown for the users to choose from.
5. **List available dentist time route(“/list”)** – this is the page for users to list the available dentist time. Users will key in the date (month and day) of their interest, then followed by the dentist that they want to view for their availability on that particular date
6. **Edit appointments route(“/editAppt”)** – this is the page for users to edit their existing appointment. An appointment Id need to be keyed and if this Id does not match the appointment id that the username is attached to, it will show an error message. If the appointment id matched the username, then they will need to key in the time and day they want to change to. However, this function can be used also use by admin and the admin can edit any appointment for the users.
7. **Browse for dentist appointments route(“/browse”)** – this is the page for admin to browse a particular dentist appointment. The dentist name will be keyed in and all the appointment for that dentist will be shown.
8. **Search for appointment route(“/search”)** – this is the page for admin to bsearch a particular appointment by appointment id. An appointment id will be keyed in and the appointment details include the date, time, dentist and the username of the patient will be shown.
9. **Error Handling and Concurrency Mechanism**

All the custom datatype (refer to the powerpoint slide for pictorial representation) function method will return error message if there is any error. A ‘defer’ anonymous function for panic recovery is also declared at beginning of the functions in admin.go and patient.go as well as server.go where the core part and server of the application are. Error handlings are also used in the server side where invalid input are keyed in and error message will be shown to the user on the browser on the error and how they should key in the input for the respective variable (int or string and range if there is any).

Goroutines were used when the system updates the data structure for the appointment and dentist list as these will not affect the core function of the application at the backend. Concurrency issue is prevented by only allowing the system to remove/update the timeslot when making/editing appointment by specify a buffered channel with size of 1.