

## 17625 A1 Part 1

by Yucen Xu

### Design Process

#### Activity Table

| Activity                 | Description                     | Step  | Participants                   | Endpoint                 | Sample Payload  | Sample Response  | Test_Success   | Test_Fail                             |
|--------------------------|---------------------------------|---|--------------------------------|--------------------------|---|--|--|---------------------------------------|
| <b>Add an Event</b>      | Add a new event to the calendar | User login -> Client make a post request with user ID -> Server check if user has access to add events -> Server return status code                                   | Doctors with add permission    | POST /events             | {'userID':1, 'eventName': 'event with xxx', 'time':'2022-11-05 08:00:00', 'description':'event description', 'doctorID':1, 'patientID':2} | {'status':200, 'eventID':1}  | Post request with correct parameters -> Check returned eventID is valid      | Post request with missing parameters. |
| <b>Edit an Event</b>     | Edit an existing event          | User login -> Client make a put request with user ID -> Server check if event exists -> Server check if user has access to update events -> Server return status code | Doctors with update permission | PUT /event/{eventID}     | {'userID':1, 'eventName': 'event with xxx', 'time':'2022-11-05 08:00:00', 'description':'event description', 'doctorID':1, 'patientID':2} | {'status':200}   | Put request with valid eventID -> Check the event is updated                 | Put request with invalid eventID.     |
| <b>Delete an Event</b>   | Delete an existing event        | User login -> Client make a put request with user ID -> Server check if event exists -> Server check if user has access to delete events -> Server return status code | Doctors delete with permission | DELETE /event            | {'eventID':1}   | {'status':200, 'eventID':1}  | Delete request with valid eventID -> Check the event no longer exists        | Delete request with invalid eventID.  |
| <b>Get all Events</b>    | Get all events for a given date | User login -> Client make a get request with user ID -> Server check if user has access to read events -> Server return status code and/or all events                 | Doctors with read permission   | GET /event/date={date}   |   | {'status':200, 'events': [{'eventID':1, 'eventName': 'event with xxx', 'time':'2022-11-05 08:00:00', 'description':'event description', 'doctorID':1, 'patientID':2}]} | Add a new event -> Get request with valid date -> Check the new event exists | Get request with invalid date.        |
| <b>Get Current Date</b>  | Get current date                | Client make a get request with user ID -> Server status code and/or current date  | All users                      | Get /date/user={userID}  |   | {'status':200, 'date':'2022-11-05'}  | Get request with valid userID -> Check the date is correct                   | Get request with invalid userID.      |
| <b>Get Current Day</b>   | Get current day                 | Client make a get request with user ID -> Server status code and/or current day   | All users                      | Get /day/user={userID}   |   | {'status':200, 'day':'05'}   | Get request with valid userID -> Check the day is correct                    | Get request with invalid userID.      |
| <b>Get Current Month</b> | Get current month               | Client make a get request with user ID -> Server status code and/or current month   | All users                      | Get /month/user={userID} |   | {'status':200, 'month':'11'}   | Get request with valid userID -> Check the month is correct                  | Get request with invalid userID.      |

| Activity               | Description         | Step   | Participants | Endpoint                | Sample Payload | Sample Response                  | Test_Success  | Test_Fail                              |
|------------------------|---------------------|--|--------------|-------------------------|----------------|----------------------------------|---|--|
| Get<br>Current<br>Year | Get current<br>year | Client make a get<br>request with user ID<br>-> Server status<br>code and/or current<br>year | All users    | Get /year/user={userID} |                | {'status':200,<br>'year':'2022'} | Get request<br>with valid<br>userID -><br>Check the<br>year is<br>correct | Get request<br>with invalid<br>userID. |