Meeting Minutes
27/08/21

Attendees: Walter, Shang Zhe, Parit, Yujian

Meeting starts 2:33pm

Shang Zhe begins by alerting the team to an oversight in DB schema design. Storing cards with images (approx 1MB in capacity) on each User will then only allow at most 16 cards. The solution is addressed in https://www.mongodb.com/basics/embedded-mongodb where it is better to store cards in a separate collection with RF keys to the user they belong, rather than storing references on users that will grow in size.

Walter asks on character escaping and about validating input. Shang Zhe explains that the process needs to be done manually for each field. He also reminds that the user will not be in the request body. Escaping input will be done through both Mongoose and React.

Parit explains his code structure in alignment to the following video guide https://www.youtube.com/watch?v=lNqaQ0wEeAo. He asks about where to structure his endpoint code, to which Shang Zhe shows where his endpoint code goes.

Shang Zhe advises that mongoose.Schema.Types.ObjectId is the type used to define an ObjectId type for the requests and that mongoose.Types.ObjectId is used for the interface definition.

The `declare` keyword can be used to declare a variable, not defining it: like C.

A word on mocking save(). Within an async function save can be awaited, this will return the result or throw an error if failed. Because express and async routes have a rocky interface, Shang Zhe has implemented a quick adaptor function to handle this, where by caught exceptions from the promises will be send to the next() function.

Richard's upload image route will be the same as card update route. The api documentation would not have to be changed in light of the new information. He requests Walter for a takeover of the branch cardRESTfulApi in order to add the details. He will wait for the updated schema to be merged to master, then will pull for development. Walter to update the Schema.

Shang Zhe adds a note for Unit Testing, mocking can include the router itself, where by DeepPartial<T> can be used to deeply make an interface's properties optional.

Once Shang Zhe finishes the authentication middleware and Walter updates the database Schema as defined here. All the endpoints should be implemented.

Regarding Sunday's deadline for an inception list, Walter proposes they create an Architectural diagram for the

system, in order to capture the system. Walter will model the system and also look for other ways to complete the inception checklist.

Meeting ended 3:49pm.