

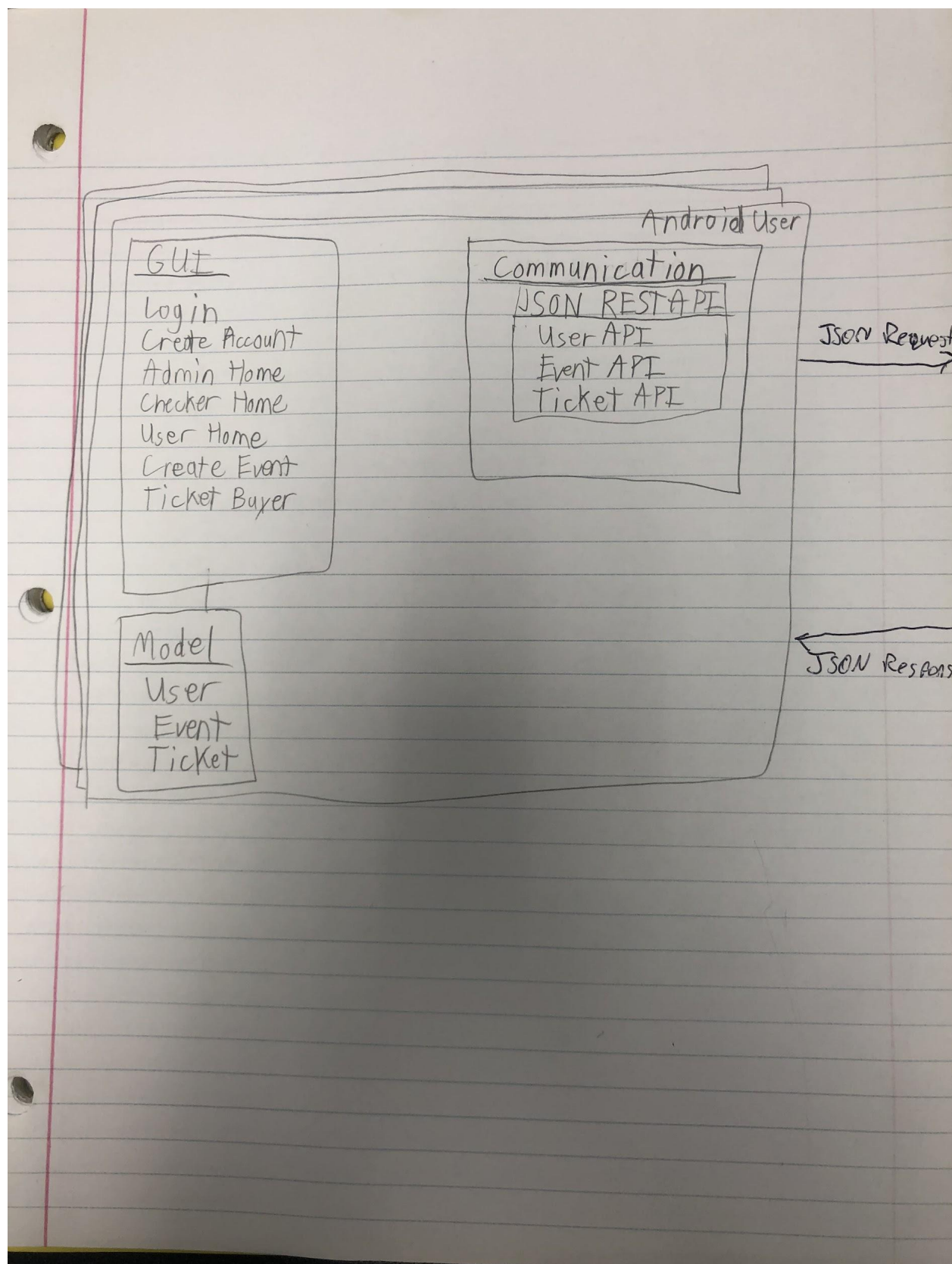
Design Document for CyTix

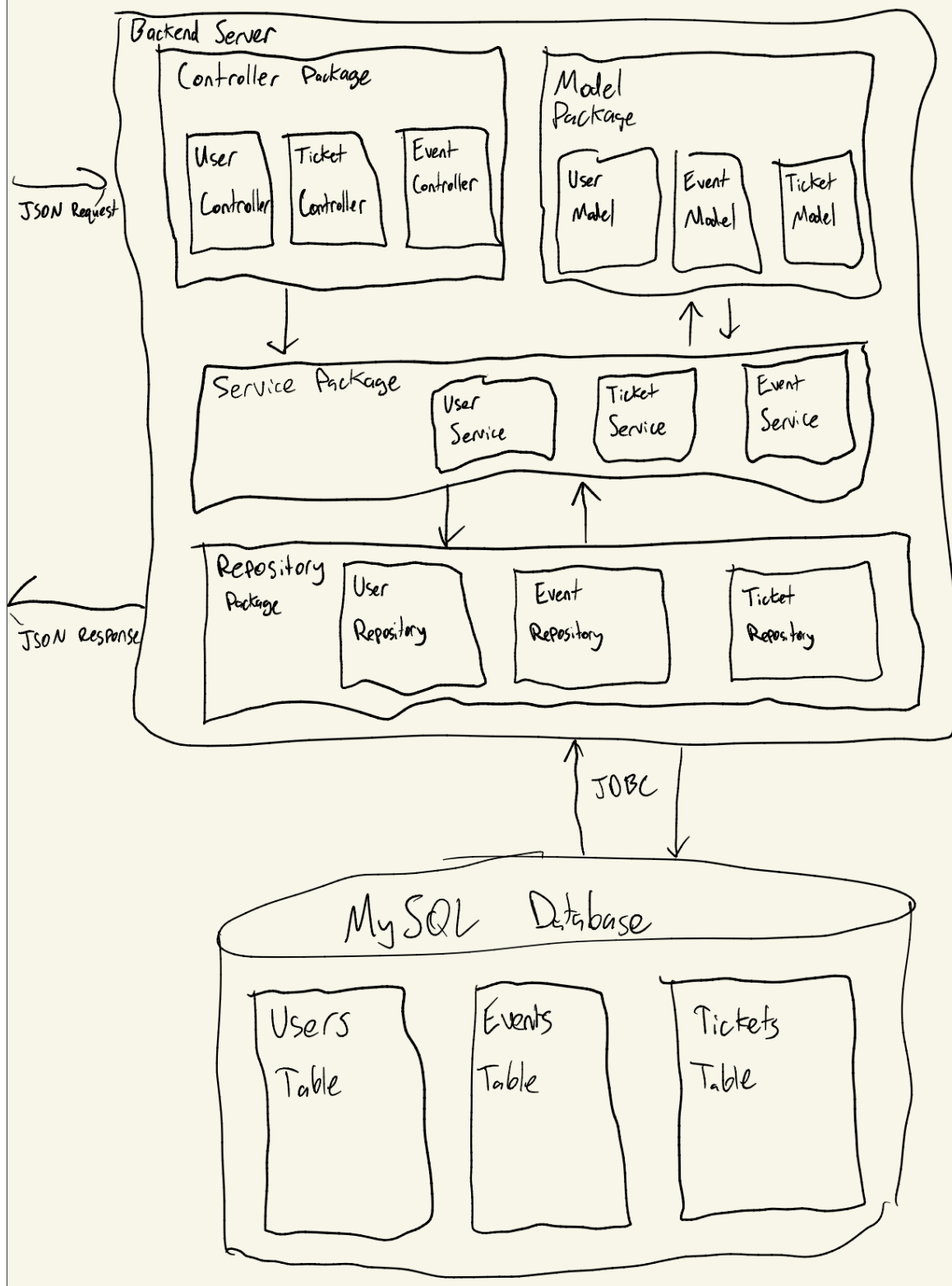
Group 1_CW_2

Ben Johnson

Jacob Larson

Jason Guo



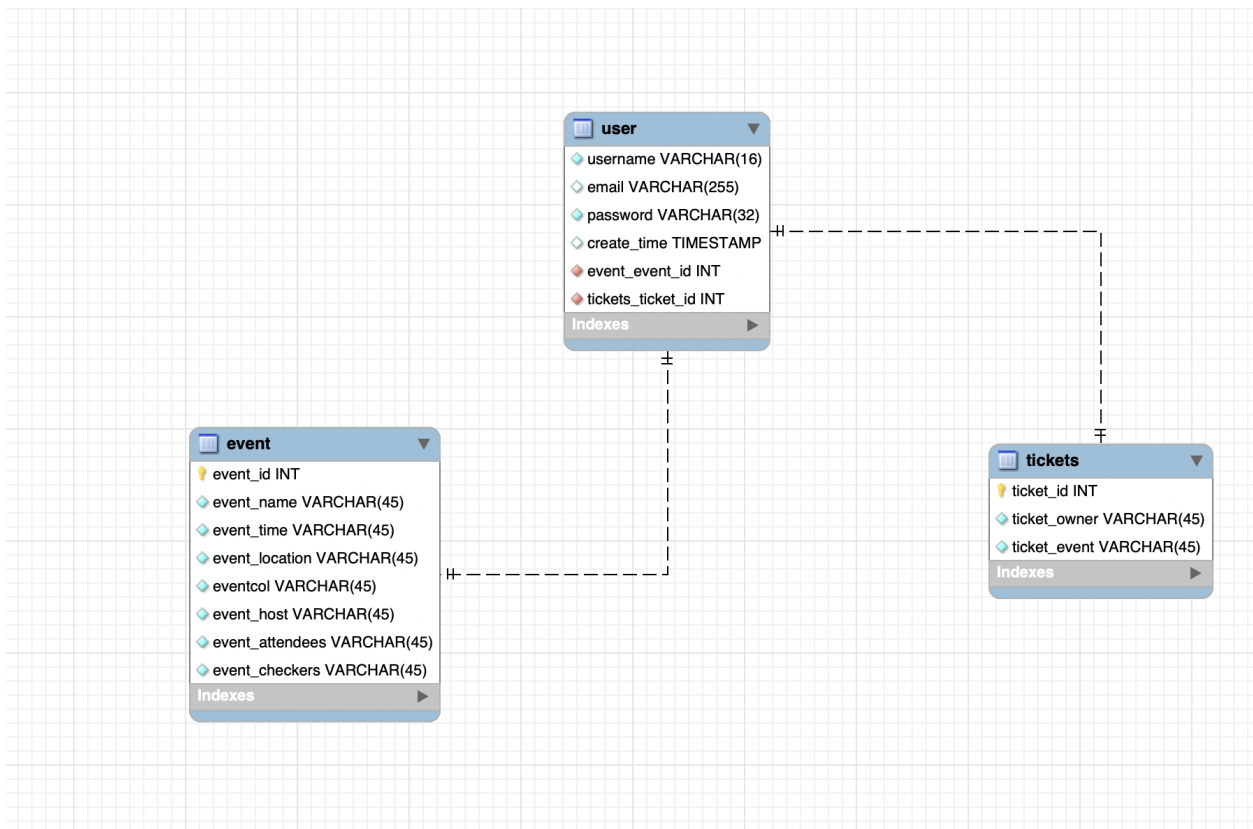


Design Description:

On the client side, the GUI allows application users to create an account and manage tickets and events. Depending on the type of account, users can also create events and tickets. The permissions for ticket and event creation are checked on the client side. When the client selects an action, an HTTP request is created and sent to the server using the REST API from the backend. Requests and responses are sent and received as JSON objects.

On the server side, User, Event, and Ticket controllers are used to manage the API endpoints. When an HTTP request is received, if it is a valid request, the controller will call a service class. There is a service class for each type of controller. The service class is what holds the program logic. The service class will perform the specified action, interacting with the respective model depending on the request type. After this is completed, the service class will then call the repository, which interacts with the database. From there, the repository class will update the database accordingly, and the server will send the proper response type depending on the type of request that was received.

The tables in the database hold the information for each user, event, and ticket. The repository class interacts with each table as specified by the service class.



Tables and Fields:

User Table: holds the username, email, password, event id, and ticket id of each user. The ticket id has a one to one relationship with the user, as each ticket has only one user. The event id has a one to one relationship with the user as well.

Event Table: holds the event name, event time, event location, event host, event attendees, and event checkers

Ticket Table: holds the ticket id, ticket owner, and ticket event information.