Design Document for CyCoach

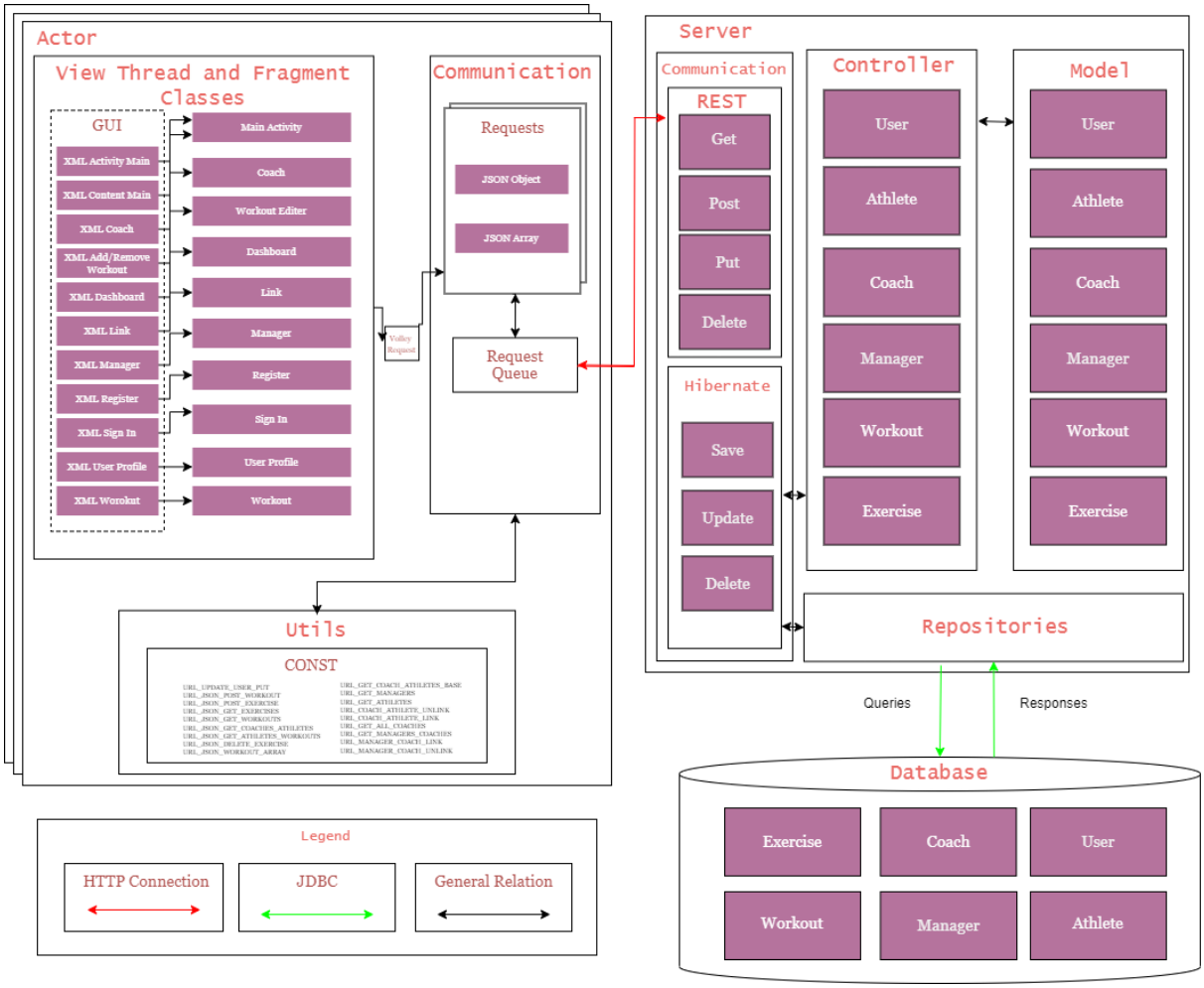
Group <UG-03>

Richard Bach 30% contribution

Zach Josten 20% contribution

Zane Eason: 25% contribution

Jayden Luse: 25% contribution


Actor
View Thread and Fragment Classes
GUI
XML Activity Main
XML Content Main
XML Coach
XML Add/Remove Workout
XML Dashboard
XML Link
XML Manager
XML Register
XML Sign In
XML User Profile
XML Workout
Main Activity
Coach
Workout Editer
Dashboard
Link
Manager
Register
Sign In
User Profile
Workout
Communication
Requests
JSON Object
JSON Array
Volley Request
Request Queue
Utils
CONST
Server
Communication
REST
Get
Post
Put
Delete
Hibernate
Save
Update
Delete
Controller
User
Athlete
Coach
Manager
Workout
Exercise
Model
User
Athlete
Coach
Manager
Workout
Exercise
Repositories
Queries
Responses
Database
Exercise
Coach
User
Workout
Manager
Athlete
Legend
HTTP Connection
JDBC
General Relation

Frontend
The actor interacts with our application using android fragments. These fragments handle user input through listeners such as onClickListeners and onItemSelectedListeners. UI is managed from within these classes by android dataBindings linked to XML files. The fragments use Volley and the static URLs stored in the Const helper class in the Utils package to handle JSONObject and JSONArray server GET, POST, PUT, DELETE requests, which are then added to the request queue. When the server responds, the classes wait for the asynchronous requests and call the success or failure methods contained in the VolleyCallback interface. An example is the sign in fragment, which gets a click from the user and checks if the user is contained in a list of users returned from the server by a GET request. If the user is found, the fragment navigates to the dashboard fragment.

Backend
Communication
The backend uses mappings to update the database based on information sent to the given mappings' URLs. These include:

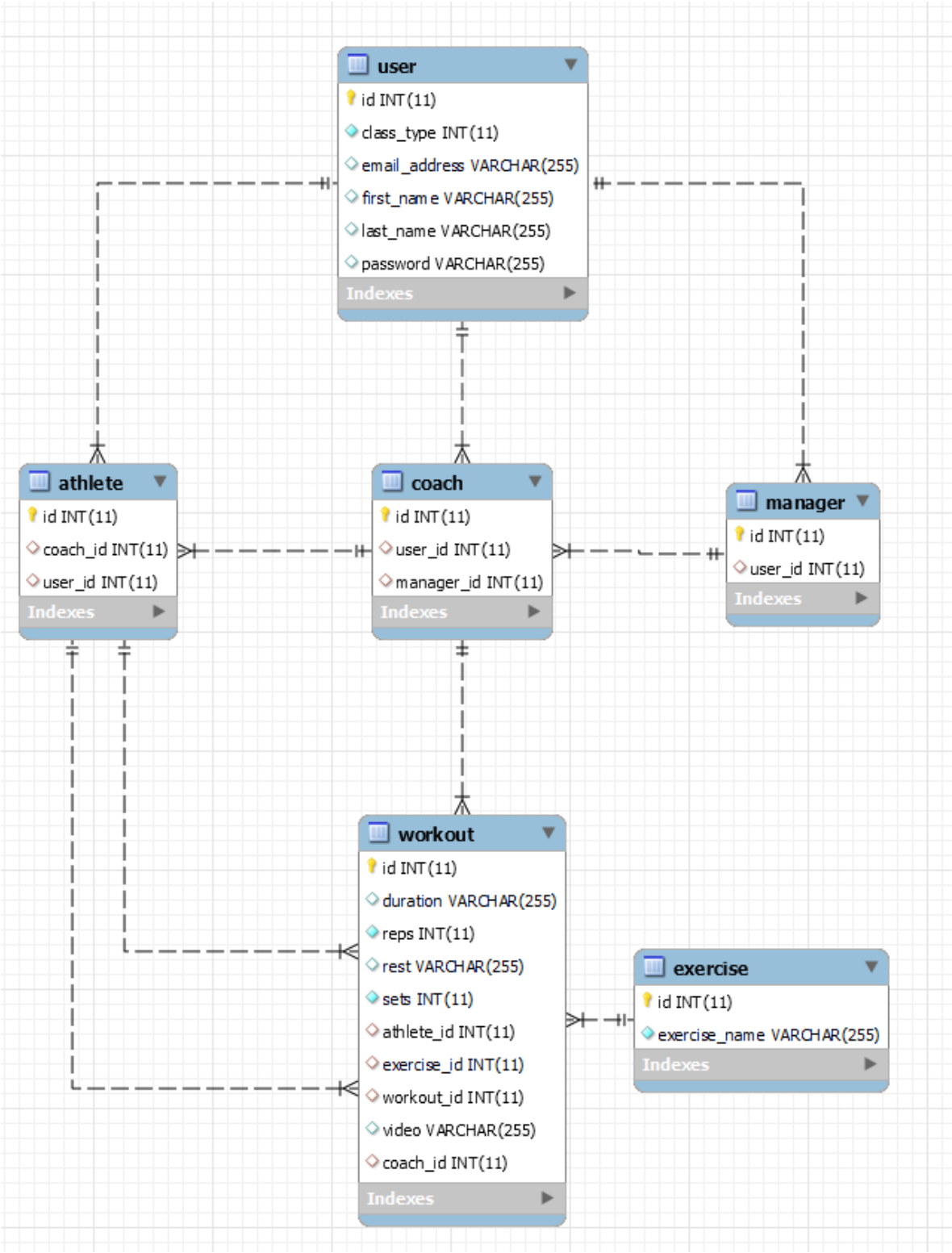
- Get: request information. Often paired with a URL parameter in order to narrow down the information returned.
- Post: push information to the server to be stored in the database. Often paired with a request body to structure the data saved to the database.
- Put: alter/update information that is already in the database. Paired with a request body to specify the changes made to the database.
- Delete: remove information from within the database.

Controllers
The controllers are used by the frontend to contact the backend in order to make changes to the database. These include:

- User: Contains the above mappings to create users, which contain one-to-one relationships with one of the following: Athlete, Coach, and Manager. A User can only have a relationship with one of these classes.
- Athlete: Contains only Get mappings. Has a one-to-one relationship with Users. Has a one-to-many relationship with Workouts representing the list of workouts an athlete is assigned by their coach.
- Coach: Contains all of the above mappings. Has a one-to-one relationship with Users, a one-to-many relationship with Athletes representing the athletes assigned to each coach, a one-to-many relationship with Workouts representing the workouts a coach has created, and a many-to-one relationship with Managers representing the manager each coach is assigned to.
- Manager: Contains all of the above mappings. Has a one-to-one relationship with Users and a one-to-many relationship with Coaches representing the list of coaches assigned to each manager.
- Workout: Contains all of the above workouts. Has a many-to-one relationship with Exercises representing how multiple workouts can be made from one exercise, a many-to-one relationship with athletes representing how one athlete can have a list of

workouts, a many-to-one relationship with Coaches representing how a coach can create many workouts.

- Exercise: Contains all of the above mappings. Has a one-to-many relationship with Workouts representing how many workouts can be made from a single exercise.


user
id INT(11)
class_type INT(11)
email_address VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
password VARCHAR(255)
Indexes
athlete
id INT(11)
coach_id INT(11)
user_id INT(11)
Indexes
coach
id INT(11)
user_id INT(11)
manager_id INT(11)
Indexes
manager
id INT(11)
user_id INT(11)
Indexes
workout
id INT(11)
duration VARCHAR(255)
reps INT(11)
rest VARCHAR(255)
sets INT(11)
athlete_id INT(11)
exercise_id INT(11)
workout_id INT(11)
video VARCHAR(255)
coach_id INT(11)
Indexes
exercise
id INT(11)
exercise_name VARCHAR(255)
Indexes