Design Document for GymApp

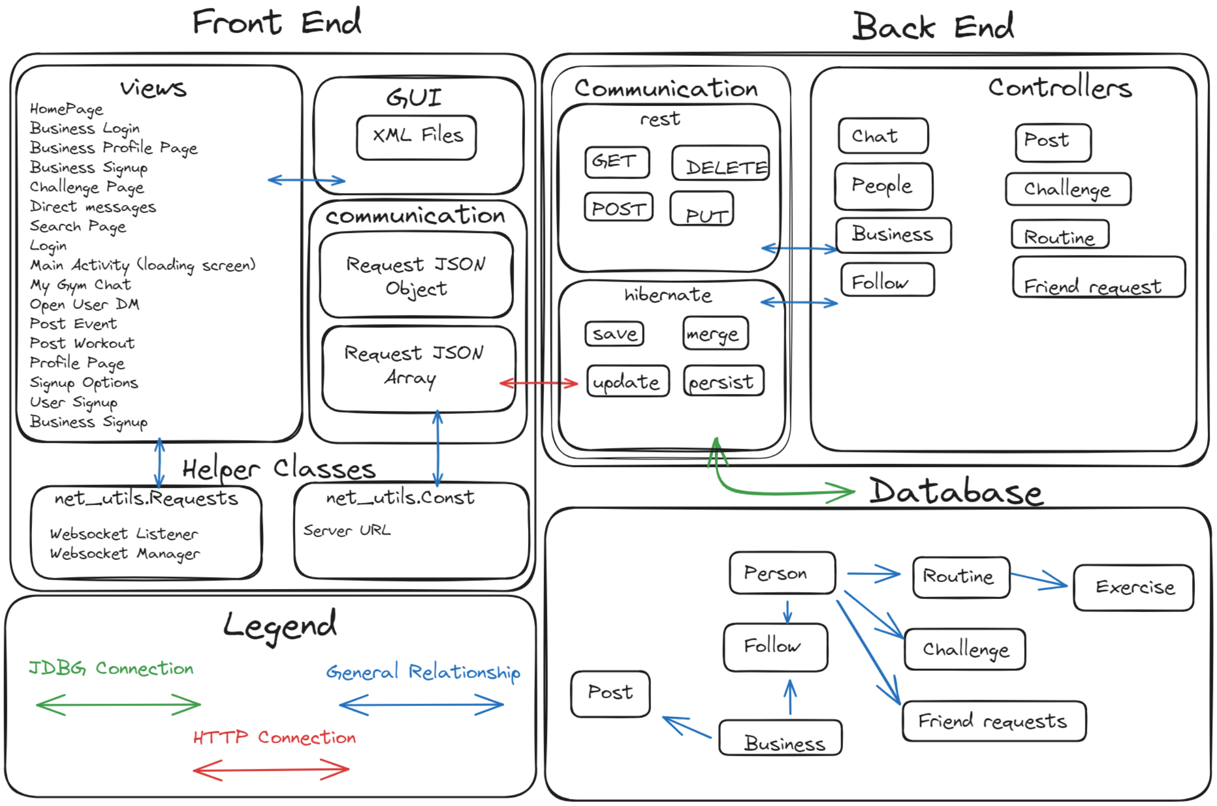
Group TA_126

Joe : 25% contribution

Kyle: 25% contribution

Lindsey Wessel: 25% contribution

Mason : 25% contribution


Front End
Views
HomePage
Business Login
Business Profile Page
Business Signup
Challenge Page
Direct messages
Search Page
Login
Main Activity (loading screen)
My Gym Chat
Open User DM
Post Event
Post Workout
Profile Page
Signup Options
User Signup
Business Signup
GUI
XML Files
communication
Request JSON Object
Request JSON Array
Helper Classes
net_utils.Requests
Websocket Listener
Websocket Manager
net_utils.Const
Server URL
Back End
Communication
rest
GET
DELETE
POST
PUT
hibernate
save
merge
update
persist
Controllers
Chat
People
Business
Follow
Post
Challenge
Routine
Friend request
Database
Person
Routine
Exercise
Follow
Challenge
Friend requests
Post
Business
Legend
JDBG Connection
General Relationship
HTTP Connection

Frontend

Challenges: A user types in another user's username and sends a challenge. All current and pending challenges are listed with the ability to accept, decline, or complete the challenge. Two buttons connected to the websocket allow the user to !encourage or !trash talk users they have challenged.

Direct Messages: A user types in another user's username and then connects to a private messaging screen between the two users. This uses a *WebSocket*. Connecting to a DM that has had previous messages, on connect, the user's message history is displayed.

My Gym Chat (Group Chats): A user can join their home gym group chat, and a business can join their group chat. This uses a *WebSocket*. On Connect, the message history for that group chat is displayed.

Business Profile Page: Loads all business information to the screen. Ability to post an event to the page, as well as view all events. A business joins their respective group chat from this screen.

User Profile Page: Users can view their current routines and previous posts, as well as their name and gym name. They can enter their gym's group chat, home screen, or followers list.

Workout Tracking Page: Users can enter PRs, sets, and reps of workouts they choose. Click the "Track Workout" button, and it will store their workout on the user's profile, also displayed on the profile page.

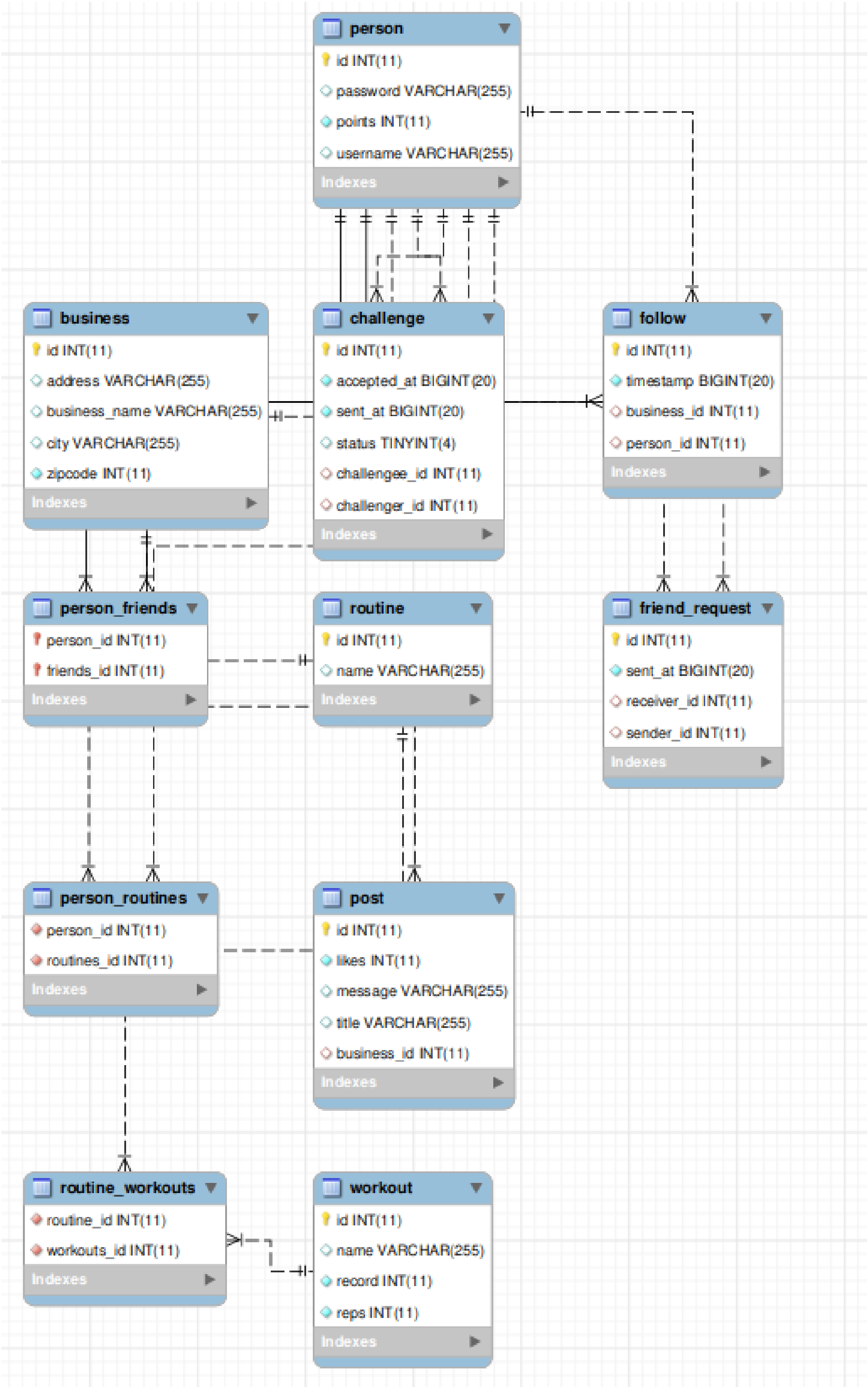
Backend

Communication

-uses REST API to send & receive information
-uses websockets to chat with one or many users

Controllers

- **Chat**: chat directly with another person or with a group of people via WebSockets
- **People**: GET, POST a person. GET a person's friends. GET people by search query, GET people by search query whose `username` is not friends with
- **Businesses:** GET, POST a business. GET businesses near a certain zip code
- **Follows**: GET, POST a follow relationship. GET all businesses a person is following. GET all people who are following a business
- **Posts:** GET, POST a business post. GET all business posts
- **Challenges:** GET, POST a challenge between two people. GET challenges from a `username`, or GET challenges to a `username`. DELETE a challenge by declining or completing it. PUT a challenge when accepting a challenge; changes status from PENDING to IN_PROGRESS
- **Routines:** GET, POST routines by a user. GET, POST workouts to/from a routine
- **Friend requests:** GET, POST a `friend request`


person
id INT(11)
password VARCHAR(255)
points INT(11)
username VARCHAR(255)
Indexes
business
id INT(11)
address VARCHAR(255)
business_name VARCHAR(255)
city VARCHAR(255)
zipcode INT(11)
Indexes
challenge
id INT(11)
accepted_at BIGINT(20)
sent_at BIGINT(20)
status TINYINT(4)
challengee_id INT(11)
challenger_id INT(11)
Indexes
follow
id INT(11)
timestamp BIGINT(20)
business_id INT(11)
person_id INT(11)
Indexes
person_friends
person_id INT(11)
friends_id INT(11)
Indexes
routine
id INT(11)
name VARCHAR(255)
Indexes
friend_request
id INT(11)
sent_at BIGINT(20)
receiver_id INT(11)
sender_id INT(11)
Indexes
person_routines
person_id INT(11)
routines_id INT(11)
Indexes
post
id INT(11)
likes INT(11)
message VARCHAR(255)
title VARCHAR(255)
business_id INT(11)
Indexes
routine_workouts
routine_id INT(11)
workouts_id INT(11)
Indexes
workout
id INT(11)
name VARCHAR(255)
record INT(11)
reps INT(11)
Indexes