Design Document for CyLife

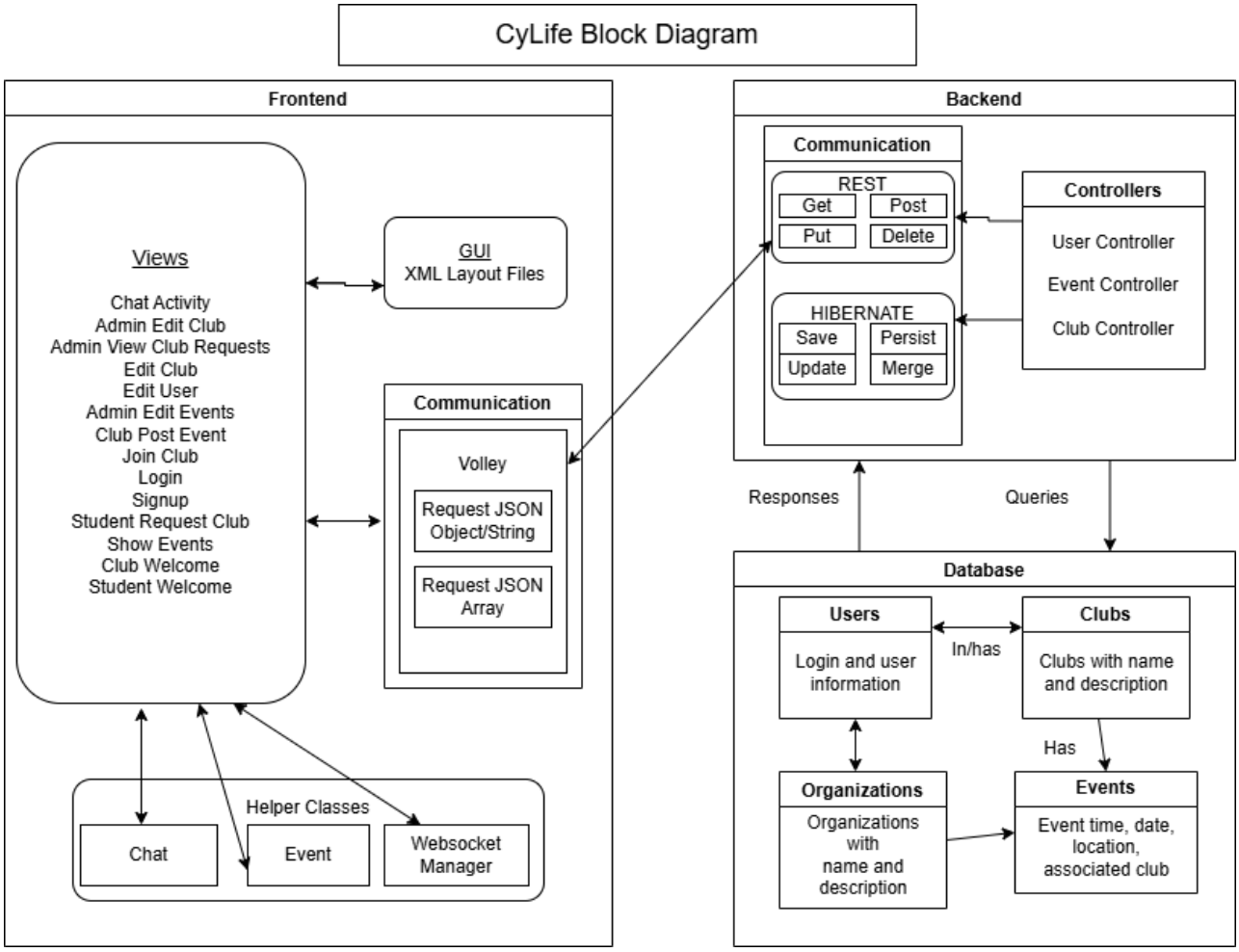
Group 4_Mahdi_2

Gregory Chernyavskiy: 25% contribution

Seth Clover: 25% contribution

Dhvani Mistry: 25% contribution

Charles Wood: 25%contribution


CyLife Block Diagram
Frontend
Views
Chat Activity
Admin Edit Club
Admin View Club Requests
Edit Club
Edit User
Admin Edit Events
Club Post Event
Join Club
Login
Signup
Student Request Club
Show Events
Club Welcome
Student Welcome
GUI
XML Layout Files
Communication
Volley
Request JSON Object/String
Request JSON Array
Helper Classes
Chat
Event
Websocket Manager
Backend
Communication
REST
Get
Post
Put
Delete
HIBERNATE
Save
Persist
Update
Merge
Controllers
User Controller
Event Controller
Club Controller
Responses
Queries
Database
Users
Login and user information
In/has
Clubs
Clubs with name and description
Has
Organizations
Organizations with name and description
Events
Event time, date, location, associated club

BLOCK DIAGRAM DESCRIPTION

FRONTEND:

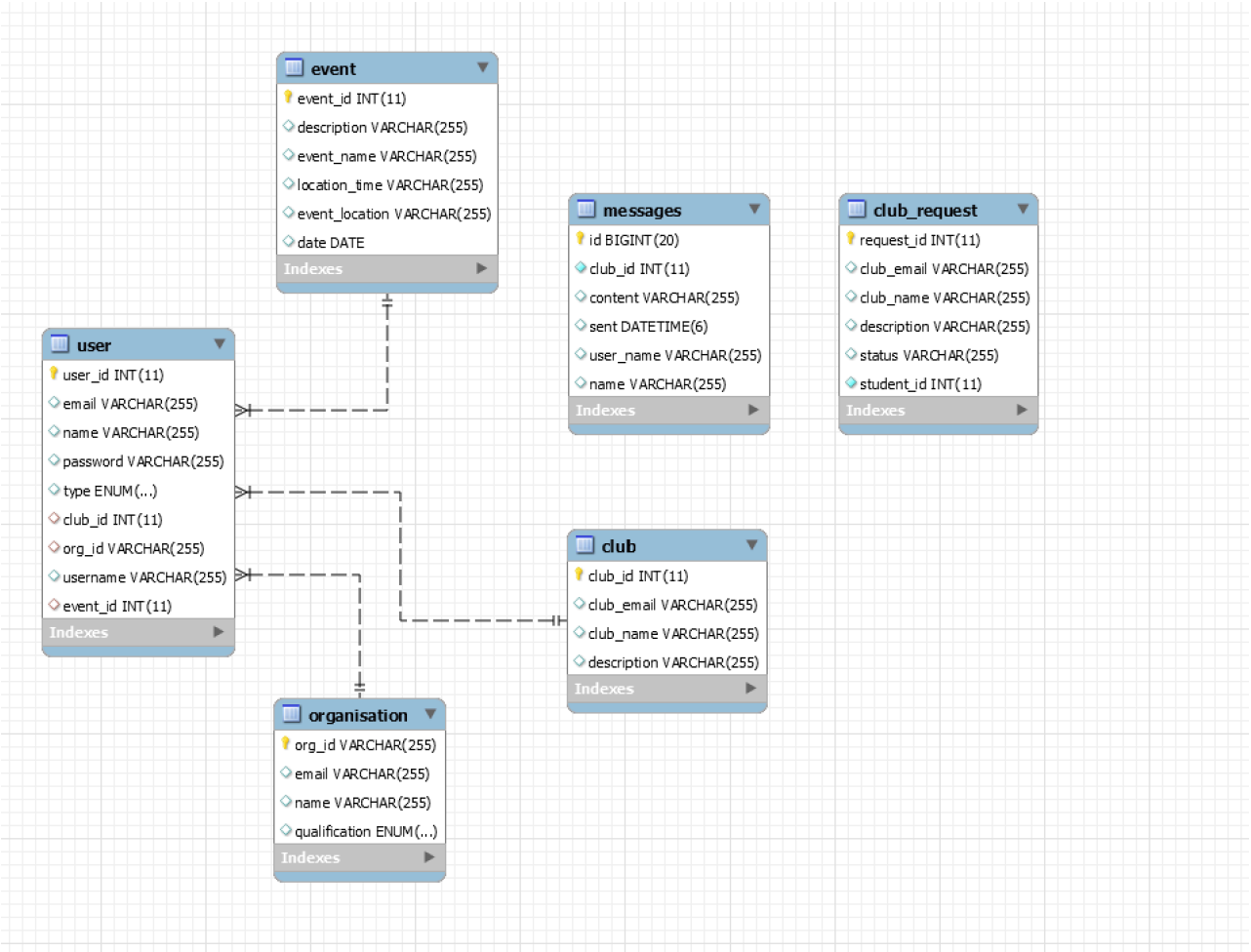
- **CreateAccount (User):**
  - Description: Provides a registration page with input fields and a submit button:
  - Fields:
    - EditText: Email (serves as the unique identifier for the user)
    - EditText: Full Name
    - EditText: Password
    - Button: CreateAccount
  - Functionality: When the CreateAccount button is clicked, the entered email and password are sent to the server in a POST request to create a new user account.
- **Description:** The login screen allows users to enter their email and password to access CyLife.
  - Fields:
    - EditText: Email
    - EditText: Password
    - Button: Login
  - Functionality: On successful login, users are directed to the main dashboard, where they can access features specific to their role (STUDENT or STAFF).
- Club/Organization Chat (User)
  - Description: Displays a chat interface for users belonging to a specific club or organization.
  - Functionality: Only users with the appropriate club/organization ID can view the chat and message history for that club. A WebSocket connection is established for real-time chat updates.
- **Notifications (User):**
  - Description: Provides notifications when a student joins a club or organization.
  - Functionality: Uses a notification WebSocket to send alerts, which include a record of past messages in the club chat.

BACKEND:

- **Communication:**
  - The backend uses REST API and WebSocket endpoints to handle data exchange and real-time updates.
    - POST: Add new data (e.g., create new accounts).
    - GET: Retrieve data (e.g., get club information).
    - WebSocket: Real-time chat and notifications.
- **Controllers:**
  - Each controller manages data flow between the frontend and the database:
  - UserController: Handles user registration, login, and access permissions based on the user type (STUDENT, CLUB, STAFF).
  - ClubController: Manages club data and relationships with users.
  - NotificationController: Sends updates when users join clubs and fetches message history for club chats.

DATABASE:

- Tables:
  - User Table: Stores user ID (database-generated), email (unique identifier), password, and user type (either STUDENT or ADMIN).
  - Clubs Table: Contains club-specific data, linked to users.
  - Organizations Table: Manages organizations and links them to clubs and events.
  - Events Table: Connected to both Clubs and Organizations, allowing them to schedule events that disappear once expired.


event
event_id INT(11)
description VARCHAR(255)
event_name VARCHAR(255)
location_time VARCHAR(255)
event_location VARCHAR(255)
date DATE
Indexes
messages
id BIGINT(20)
club_id INT(11)
content VARCHAR(255)
sent DATETIME(6)
user_name VARCHAR(255)
name VARCHAR(255)
Indexes
club_request
request_id INT(11)
club_email VARCHAR(255)
club_name VARCHAR(255)
description VARCHAR(255)
status VARCHAR(255)
student_id INT(11)
Indexes
user
user_id INT(11)
email VARCHAR(255)
name VARCHAR(255)
password VARCHAR(255)
type ENUM(...)
club_id INT(11)
org_id VARCHAR(255)
username VARCHAR(255)
event_id INT(11)
Indexes
club
club_id INT(11)
club_email VARCHAR(255)
club_name VARCHAR(255)
description VARCHAR(255)
Indexes
organisation
org_id VARCHAR(255)
email VARCHAR(255)
name VARCHAR(255)
qualification ENUM(...)
Indexes