UCF CLUB & EVENT MANAGER

OUR TEAM

Nehito Dorval	Frontend Web
Andy Nguyen	Frontend Web / Frontend App
Colin Kirby	Product Manager / Frontend Web
Tylon Robinson	API / Unit Testing
Juwel Belizaire	API / Database
Jovanny Vasquez	API / Frontend App

TECHNOLOGY USED

We utilized the MERN STACK :

M MongoDB →

- Database for storing User, Club, and Event data.

E Express →

- Handles Backend / Frontend Communication through API Endpoints.

R React →

- Building Blocks for our Frontend Development.

N Node →

- Serves as the runtime environment for our JavaScript (React)

TECHNOLOGY USED

Hosting & Collaboration Tools :

Heroku →

- Cloud Service for deploying and hosting our Web App.

GitHub →

- Served as repository for us to view each others progress and easily merge / collaborate our different codes together.

TeamGANTT →

- Organized and set deadlines for our tasks with daily feedback on our status to keep us on track.

Discord →

- Served as main modem for communication for our group, used with TeamGANTT daily to maintain workflow.

TECHNOLOGY USED

Dev & Design Tools:

TailWind →

- CSS building blocks for efficient, responsive frontend styling.

Postman →

- API development and testing tool for validating endpoint functionality.

SwaggerHub →

- Interactive platform for designing and documenting APIs.

Flutter →

- UI toolkit for implementing a mobile application from our website.

Jest →

- JavaScript testing framework for ensuring code quality and reliability.

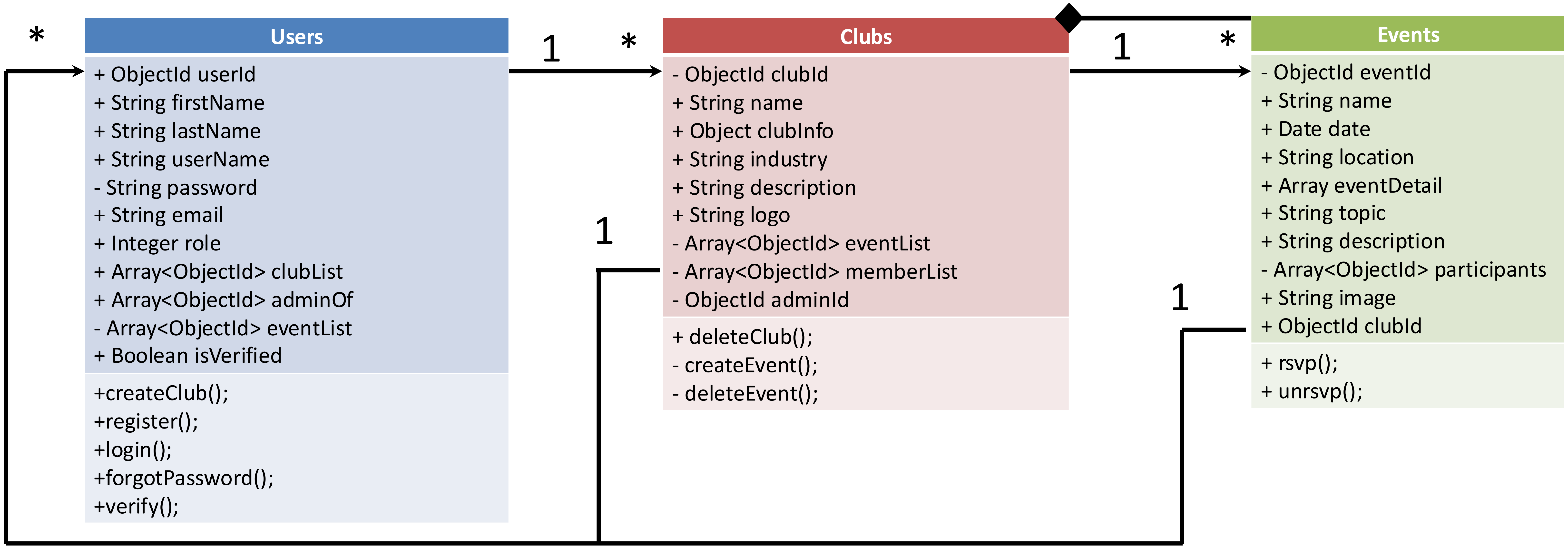
THINGS THAT WENT WELL

- Database Deployment & Implementation
  - ✓ Using MongoDB was very direct, implementing and changing the models for our data types / collections provided minimal issue throughout the course of the project.
- Using Tailwind CSS for UI Development
  - ✓ The available classes that Tailwind provided saved so much time in creating a responsive and clean interface in comparison to hard-coded CSS, and it was very simple to use by utilizing Tailwindcss.com's Component Codes.
- API Endpoint / Model Creation
  - ✓ After set up of the Database, the creation of the first models and API endpoints with their corresponding testing in Postman were very straightforward, and provided a great base to build upon, by the end of website development we only needed to add one endpoint to handle an edge case.
- Frontend Development for App
  - ✓ The App's backend was very direct with Android, as in our case it implemented the websites already working backend functions, the only needed action was the connection between the two within Flutter.

THINGS THAT DIDN'T GO WELL

- Initial Connection of Frontend to Backend Connections of Web App
  - ✓ Implementing the proper routes & controllers with the proper parameters was a chore upon the first deployment, lots of small errors inter-connected between the frontend and backend was difficult at first.
- Frontend Development for Mobile App with Dart
  - ✓ The language was very specific with a steep learning curve. Integrating with the backend was very different in comparison to the web app, required a significant adaptation from our mobile team.
- Email Implementation for Verification / Forgot Password
  - ✓ The specifics of the .env that needed to be included provided a lot of issues considering Gmail required multiple forms of authentication and extra lines of code to make sure emails are both sent and received properly.
- Functions for Unit Testing
  - ✓ Upon original planning of the backend / frontend processes of the website we didn't have too many functions to test. Figuring out how to implement the Unit Testing with our already set up web and mobile app without needing to add more functions to fit the requirement proved to be an obstacle at first.

Class Diagram



Users

+ ObjectId userId
+ String firstName
+ String lastName
+ String userName
- String password
+ String email
+ Integer role
+ Array<ObjectId> clubList
+ Array<ObjectId> adminOf
- Array<ObjectId> eventList
+ Boolean isVerified

+createClub();
+register();
+login();
+forgotPassword();
+verify();

Clubs

- ObjectId clubId
+ String name
+ Object clubInfo
+ String industry
+ String description
+ String logo
- Array<ObjectId> eventList
- Array<ObjectId> memberList
- ObjectId adminId

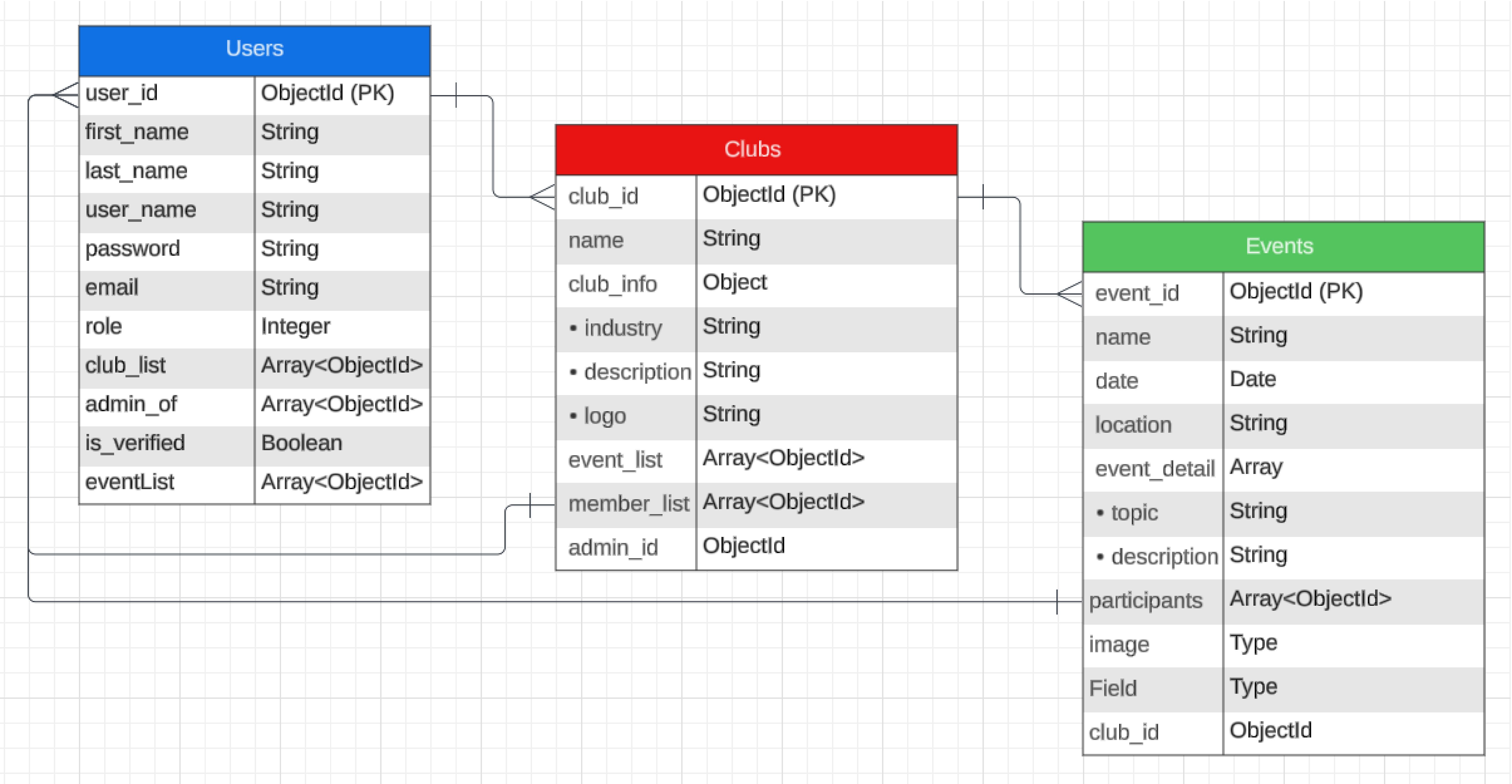
+ deleteClub();
- createEvent();
- deleteEvent();

Events

- ObjectId eventId
+ String name
+ Date date
+ String location
+ Array eventDetail
+ String topic
+ String description
- Array<ObjectId> participants
+ String image
+ ObjectId clubId

+ rsvp();
+ unrsvp();

Entity Relationship Diagram

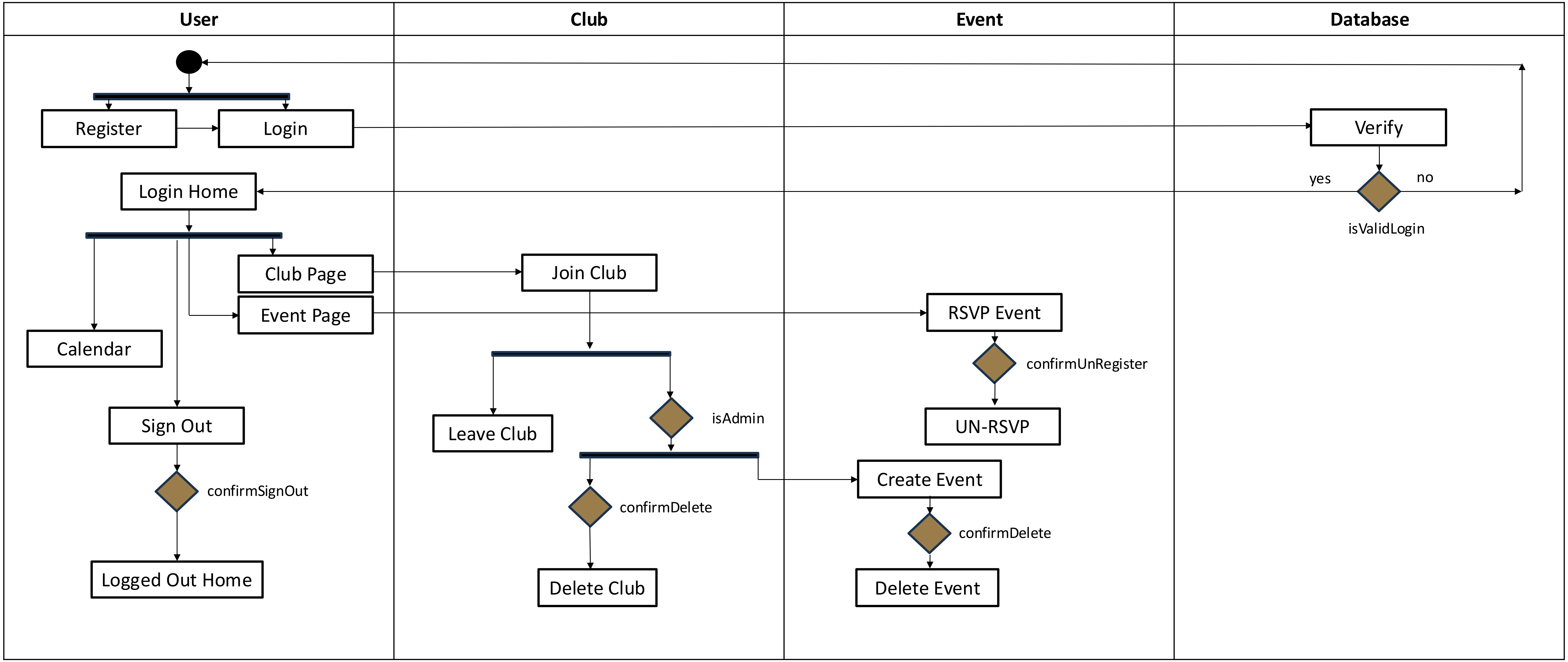


Users	
user_id	ObjectId (PK)
first_name	String
last_name	String
user_name	String
password	String
email	String
role	Integer
club_list	Array<ObjectId>
admin_of	Array<ObjectId>
is_verified	Boolean
eventList	Array<ObjectId>

Clubs	
club_id	ObjectId (PK)
name	String
club_info	Object
• industry	String
• description	String
• logo	String
event_list	Array<ObjectId>
member_list	Array<ObjectId>
admin_id	ObjectId

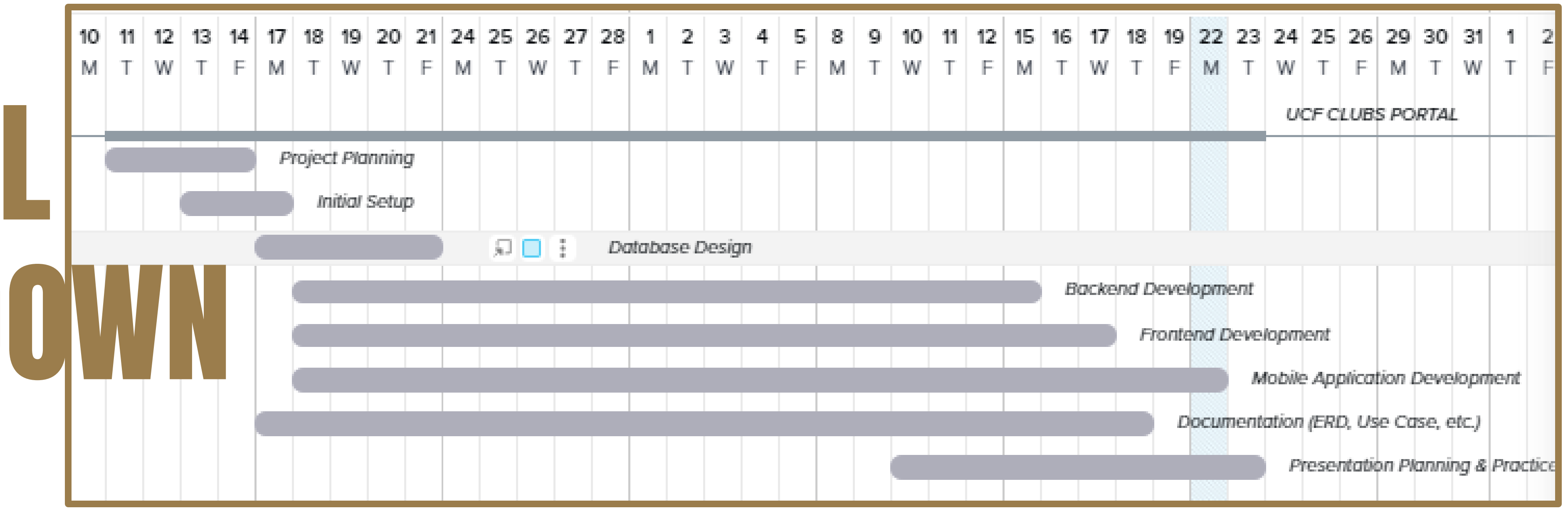
Events	
event_id	ObjectId (PK)
name	String
date	Date
location	String
event_detail	Array
• topic	String
• description	String
participants	Array<ObjectId>
image	Type
Field	Type
club_id	ObjectId

Sequence Diagram



GANTT

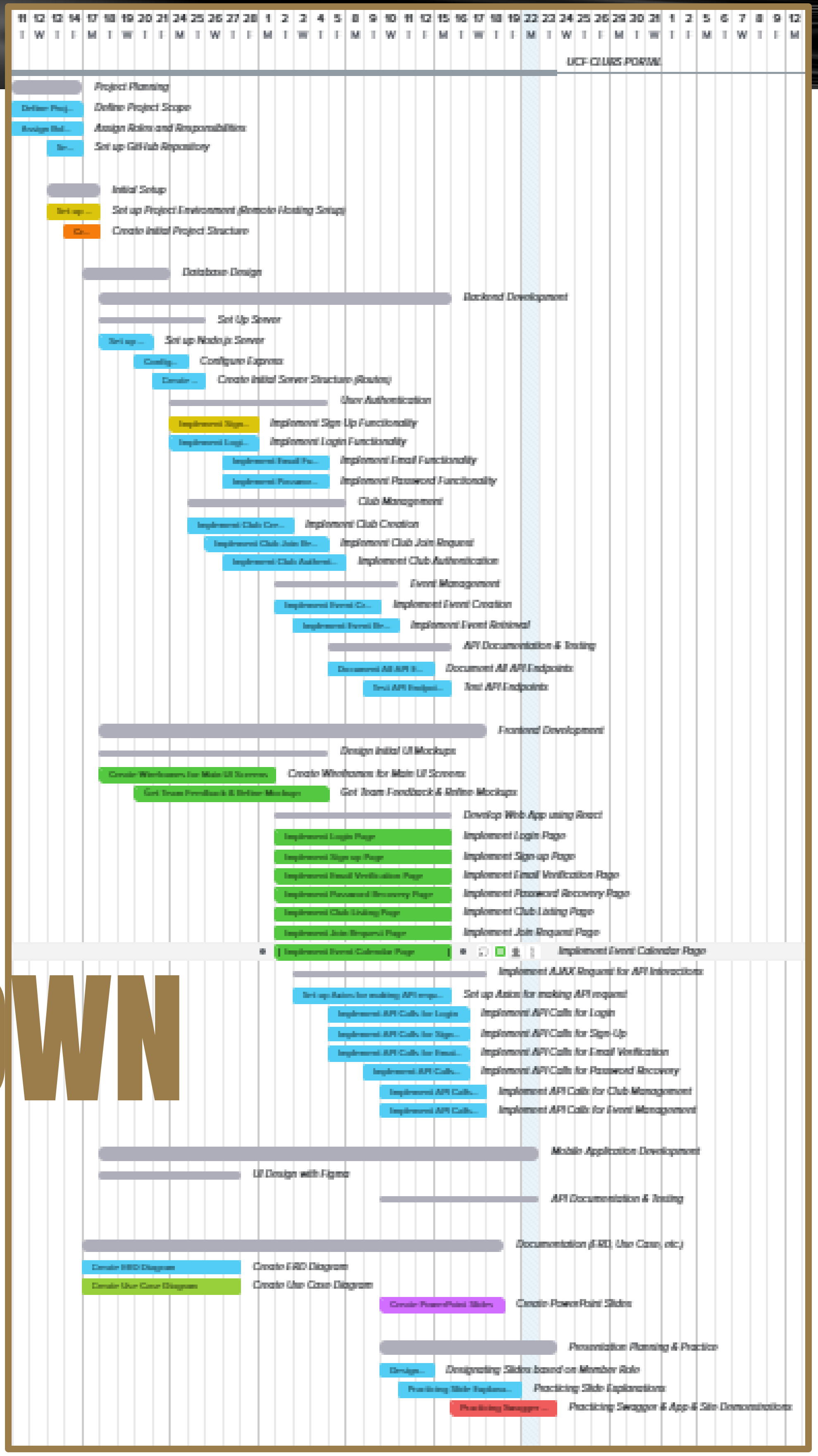
GENERAL BREAKDOWN



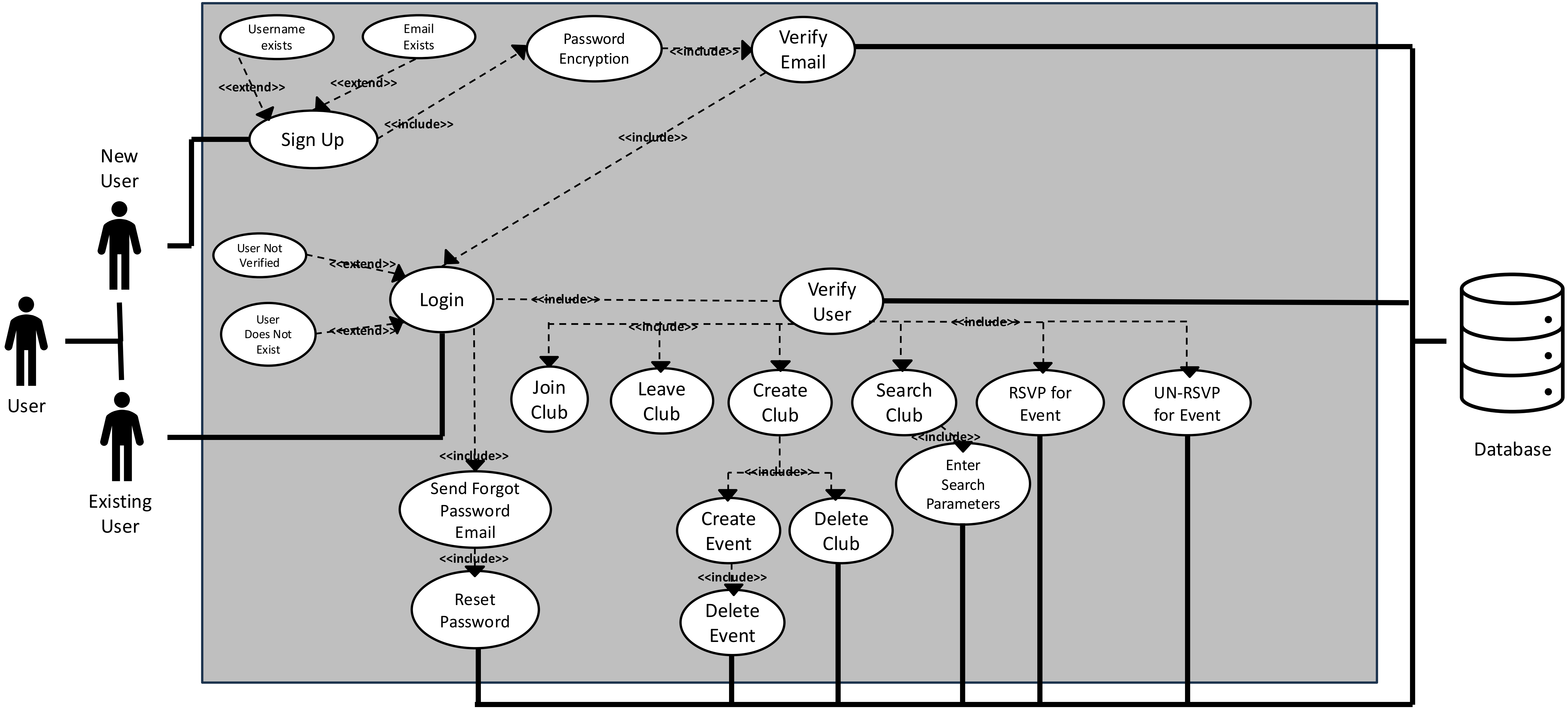
List:

- Project Planning
- Initial Set Up
- Database Design Implementation
- Backend API Development
- Frontend Design & Development
- Mobile App Development
- Documentation
- Presentation Planning

ALL INDIVIDUAL TASKS BREAKDOWN

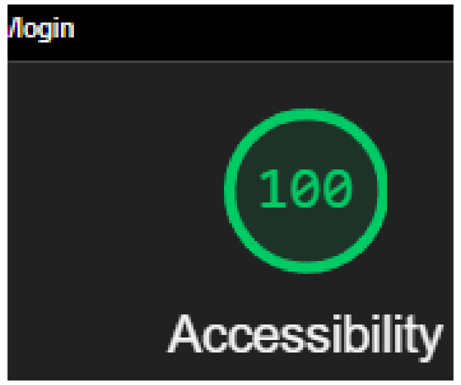


Use Case Diagram



Google Lighthouse

Login

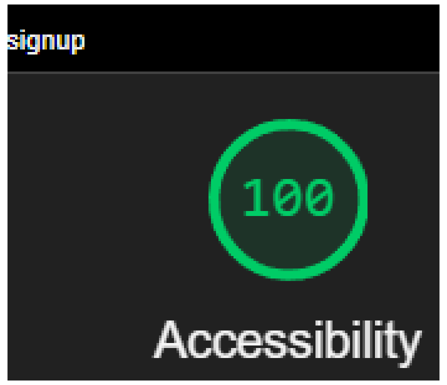


/login

100

Accessibility

Sign-up

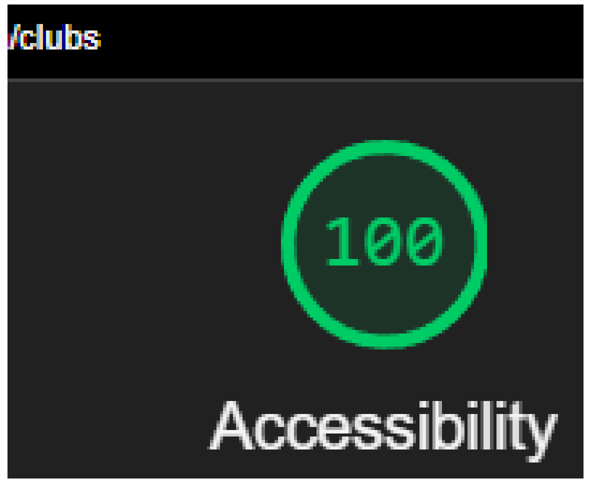


signup

100

Accessibility

Clubs

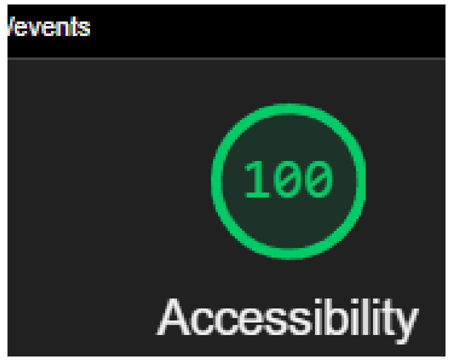


/clubs

100

Accessibility

Events

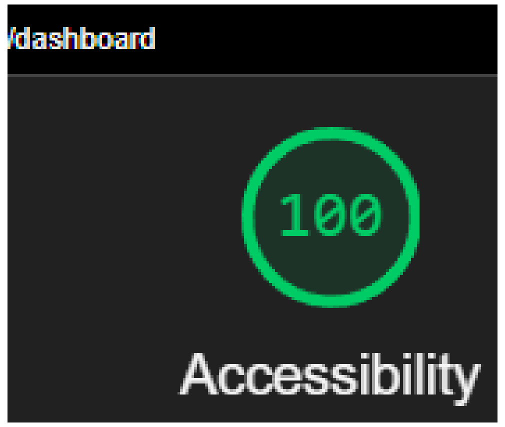


/events

100

Accessibility

Dashboard

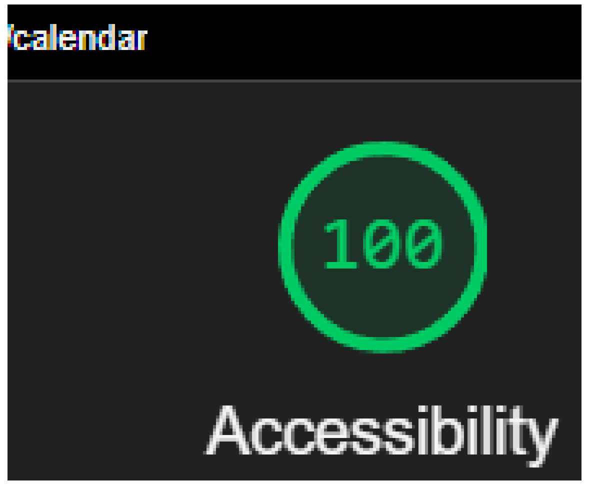


/dashboard

100

Accessibility

Calendar

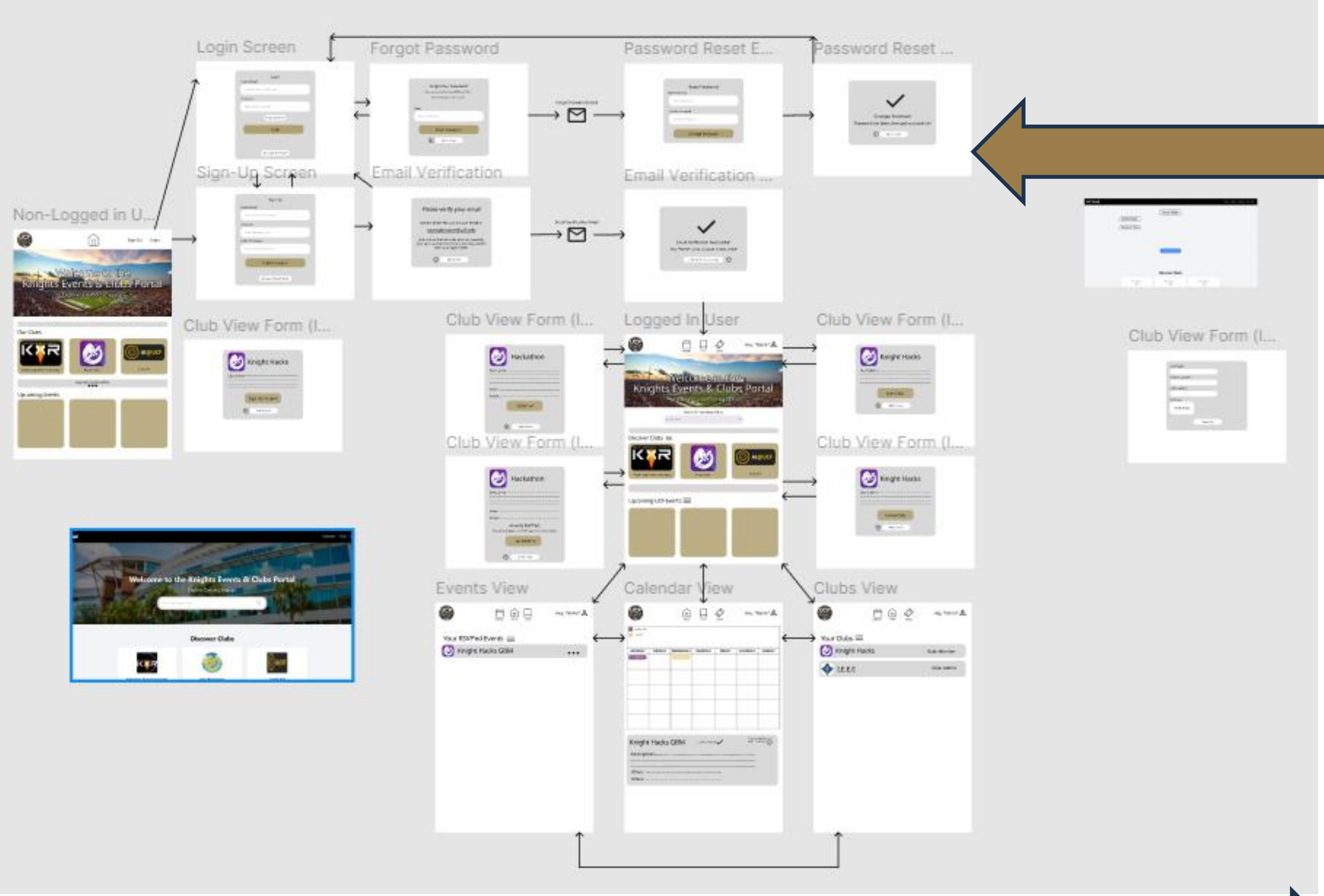


/calendar

100

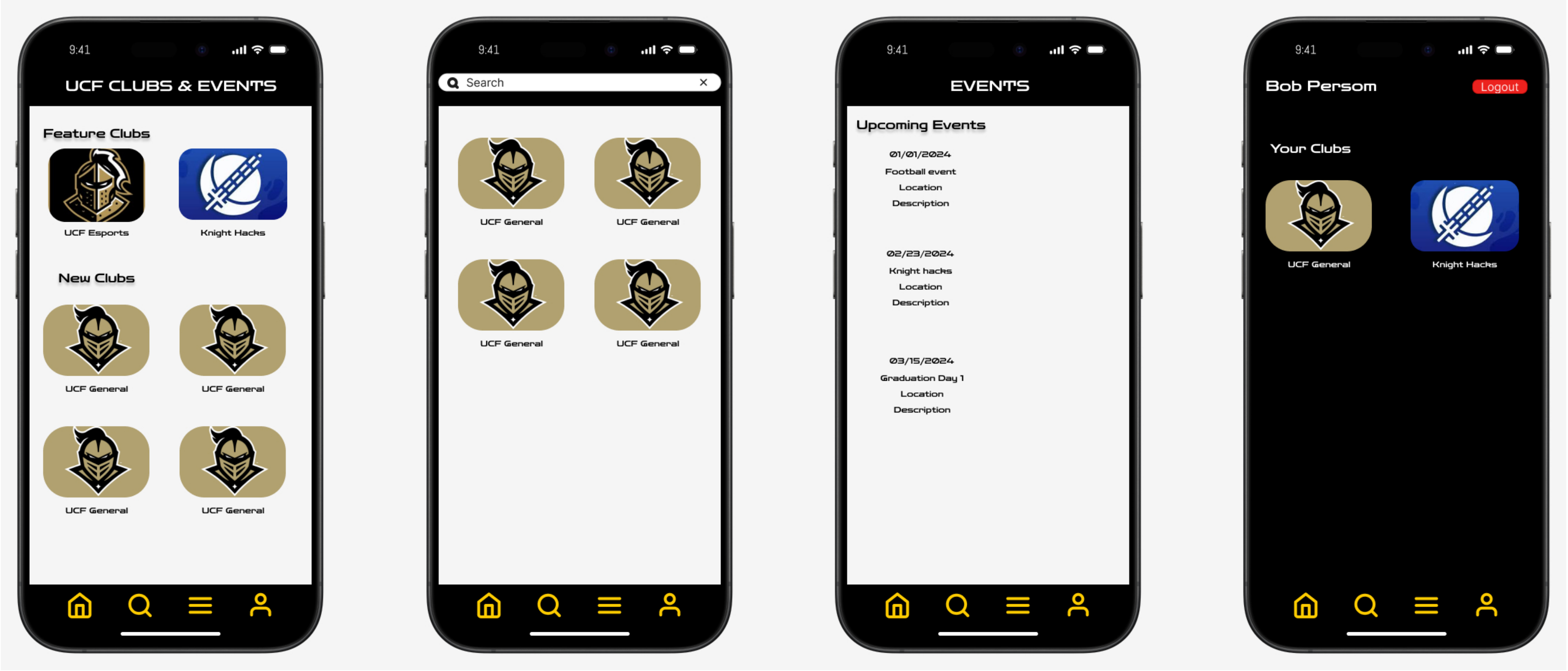
Accessibility

MOCKUPS



WEB DESIGN WIREFRAMES

APP DESIGN



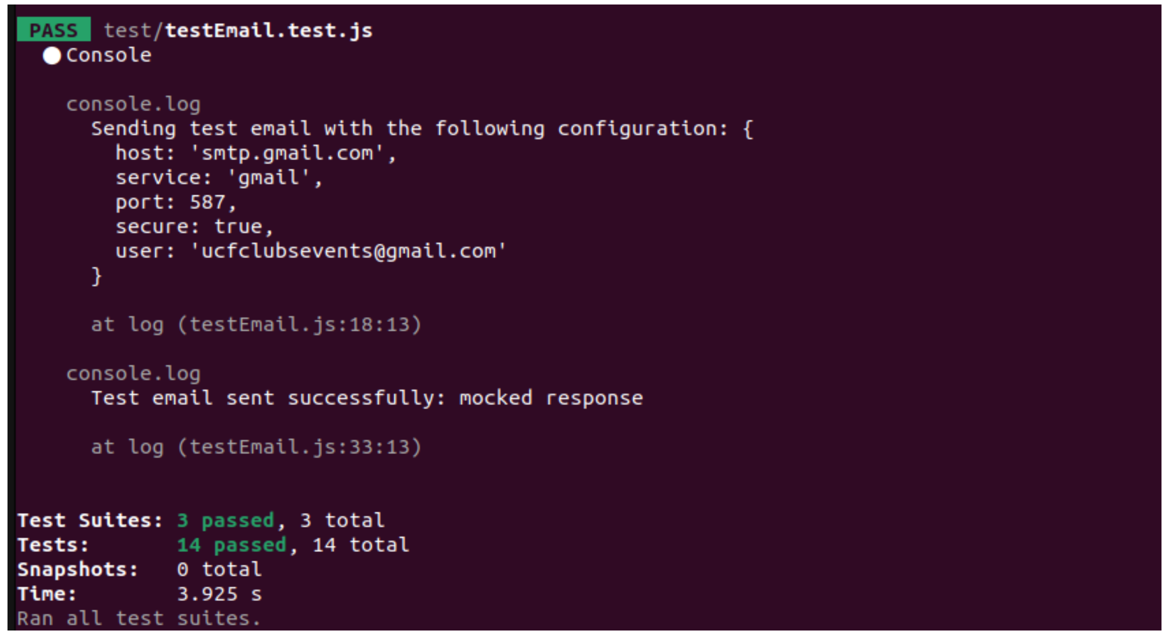
Palette

Color Palette Throughout the Website & App (Inspired by UCF Colors)

000000	NavBar, Text
F9BB0F	Hover Effects
FFCC00	Accent, Buttons
F5F5F5	Calendar, Event, Club Page Accents
FFFFFF	Background, Card Accents

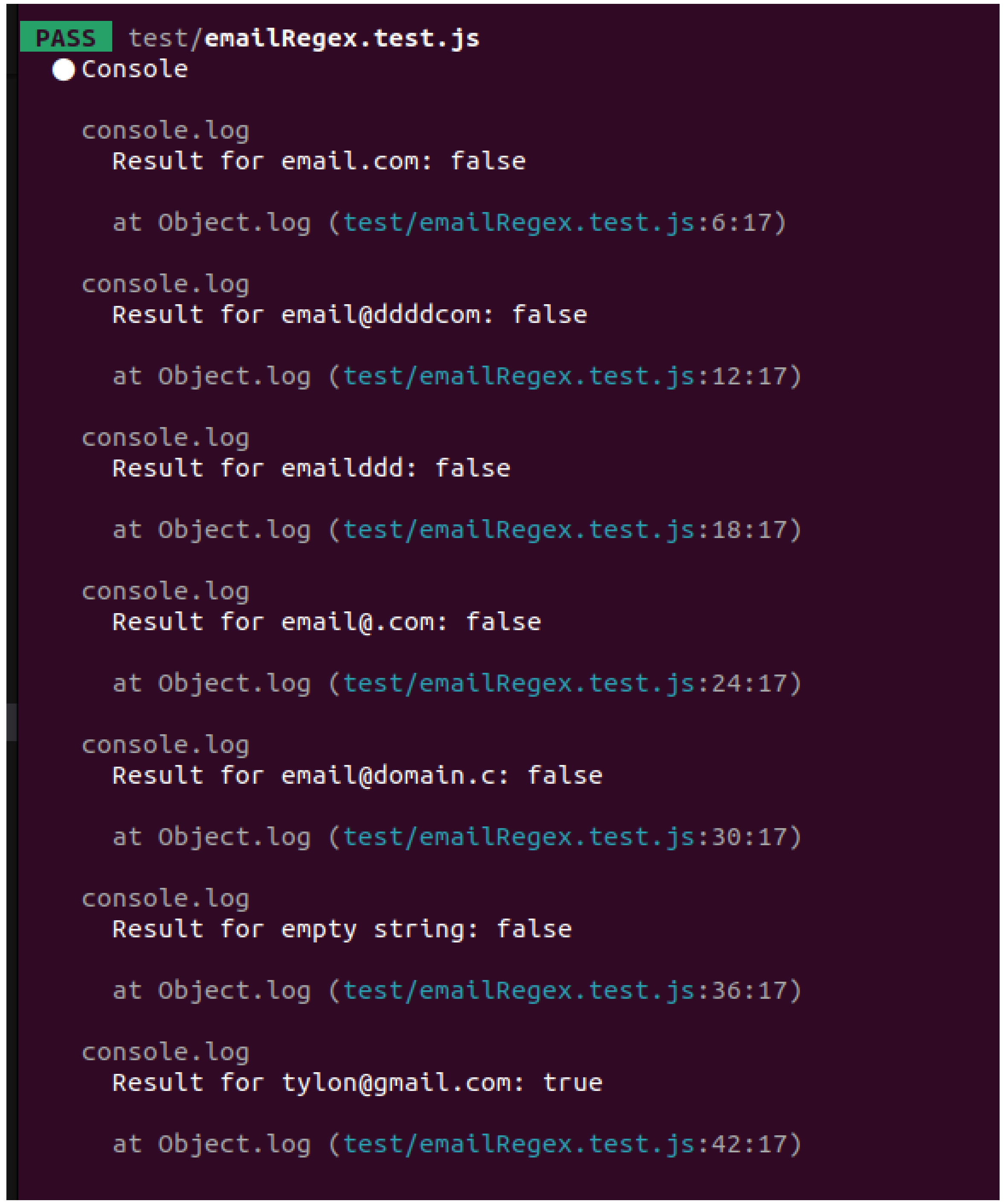
Unit Testing (Web & App)

TestEmail.test.js



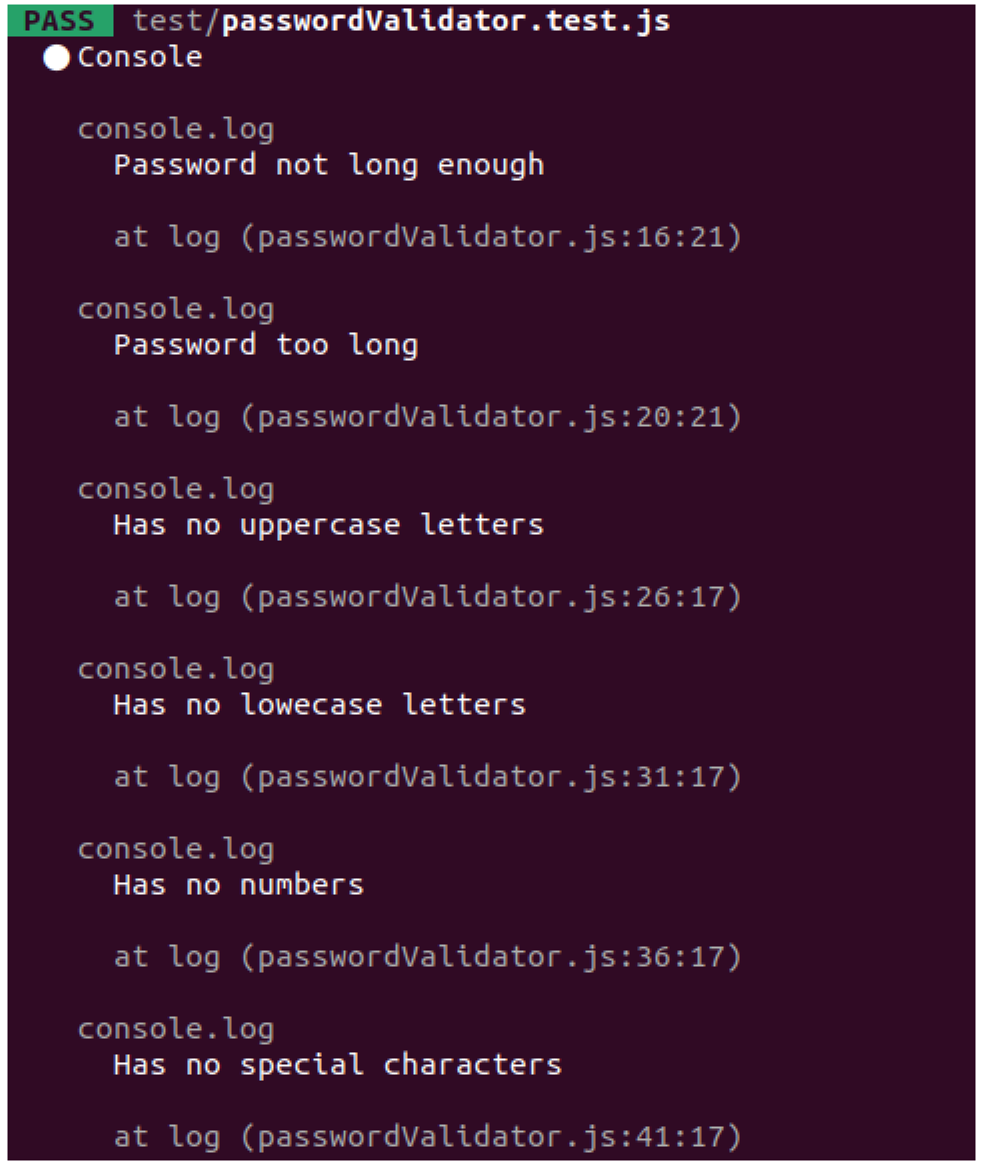
```
PASS  test/testEmail.test.js
  ● Console

    console.log
      Sending test email with the following configuration: {
        host: 'smtp.gmail.com',
        service: 'gmail',
        port: 587,
        secure: true,
        user: 'ucfclubsevents@gmail.com'
      }

      at log (testEmail.js:18:13)

    console.log
      Test email sent successfully: mocked response

      at log (testEmail.js:33:13)


Test Suites: 3 passed, 3 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        3.925 s
Ran all test suites.
```

emailRegex.test.js



```
PASS  test/emailRegex.test.js
  ● Console

    console.log
      Result for email.com: false

      at Object.log (test/emailRegex.test.js:6:17)

    console.log
      Result for email@ddddcom: false

      at Object.log (test/emailRegex.test.js:12:17)

    console.log
      Result for emailddd: false

      at Object.log (test/emailRegex.test.js:18:17)

    console.log
      Result for email@.com: false

      at Object.log (test/emailRegex.test.js:24:17)

    console.log
      Result for email@domain.c: false

      at Object.log (test/emailRegex.test.js:30:17)

    console.log
      Result for empty string: false

      at Object.log (test/emailRegex.test.js:36:17)

    console.log
      Result for tylon@gmail.com: true

      at Object.log (test/emailRegex.test.js:42:17)
```

passwordValidator.test.js



```
PASS  test/passwordValidator.test.js
  ● Console

    console.log
      Password not long enough

      at log (passwordValidator.js:16:21)

    console.log
      Password too long

      at log (passwordValidator.js:20:21)

    console.log
      Has no uppercase letters

      at log (passwordValidator.js:26:17)

    console.log
      Has no lowecase letters

      at log (passwordValidator.js:31:17)

    console.log
      Has no numbers

      at log (passwordValidator.js:36:17)

    console.log
      Has no special characters

      at log (passwordValidator.js:41:17)
```

SwaggerHub

[LargeProject | 1.0.0 | RONDOISBOSS1000_1 | SwaggerHub]

Website Demo

[UCF Club & Events Manager (ucf-club-and-event-manager-1c53fb944ab8.herokuapp.com)](ucf-club-and-event-manager-1c53fb944ab8.herokuapp.com)

Demo Agenda →

- Sign Up
- Email Verification
- Login (w/ Error Handling)
- Forgot Password
- Join / Leave Club
- RSVP / UN-RSVP
- Create Club
- Create Event
- Calendar View

App Demo

- Time for the App Demo!!!

Questions?