RUNTIME WARRIORS

System Design Document

v.Sprint3

Anabelle Hsiao · Jeremy La · Ricky Su · Mohamad El Kadri · Nevin Wong · Ivy Wills

Table of Contents

Frontend CRC

Events Page
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- Get an instance of all events from the backend - Display said events to the user in a compact view - When an event is clicked, navigate to the singular event page of that event	- Search - EventsDisplay - Event Page (singular) - user

Event Page (singular)
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- Get an instance of a specific event from the backend - Display the information of said event including name, time, date, city, etc. - Link to a Google Maps link of the event's location	- Event - user

EventsDisplay
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Given any collection of events, display them all to the user in a compact view	EventCard

EventCard
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Given an instance of a singular	n/a

event, display it to the user in a compact view	

My Events	
Parent/Subclasses: n/a	
Responsibilities ● For a user to view and update events that they created	Collaborators ● Side bar ● Event Page (singular) ● EventCard ● EventsDisplay ● Update Event can be reached from the event page singular

Attending Events	
Parent/Subclasses: n/a	
Responsibilities ● For a user to view the events that they've sign up to attend.	Collaborators ● Side bar ● Event Page (singular) ● EventCard ● EventsDisplay

Add Events	
Parent/Subclasses: n/a	
Responsibilities ● For a user to add an event to the database. ● Allows user to specify title, description, location, image, time, date, category, and whether or not the event is intended for women only	Collaborators ● Side bar ● After an event is added it will appear on the event page of all users. ● Can be added to the liked events page by any user. ● If a user attends this event it will appear in the history page

Liked
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
● Provides the logged in user a list with the events they liked	● SidebarPro ● LikesDAO

Login Page
Parent/Subclasses: n/a

Responsibilities:	Collaborators:

● If registered
   1. the page provides the user the ability to log into their account
   2. The user is able to access user-only features once logged in, such as join events,get notifications,etc
   3. The user can see their user name on the side bar
   4. The user is able to logged out
● If not registered
   1. The page is able to redirect to the register page, and the user can register
● Using JWT token, remember the logged in user
	● SidebarPro
	● User
	● Register
	● UserInfo

Register
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
● Updates global state when user clicks outside of Register. ● Updates user info state (username, password, email, firstname, lastname, gender) per keystroke. ● Posts user info when submit button is clicked.	● RegisterContext ● User

RegisterContext	
Parent/Subclasses: Context	

Responsibilities:
   ● Stores global state to be used in other classes.
	Collaborators:
	   ● n/a

SidebarPro	
Parent/Subclasses: n/a	

Responsibilities:
   ● Act as a router between the different pages of the application.
	Collaborators:
	   ● Events Page
	   ● Liked
	   ● Login
	   ● Register

Profile	
Parent/Subclasses: n/a	

Responsibilities:
   ● Fetches users current profile info into forms.
   ● Allows user to edit forms and submit to change database.
   ● Allows user to enter old password and new password. If old password is correct, changes database.
	Collaborators:
	   ● SidebarPro
	   ● localStorage

ForgotPassword	
Parent/Subclasses: n/a	

Responsibilities:
   ● Allows user to submit password reset request by submitting email.
   ● If email is valid, /forgot-password is called.
	Collaborators:
	   ● SidebarPro

ResetPassword
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- Checks url after/reset/ and checks if this token is a valid PasswordResetToken. - If not valid, displays error page. - If valid, allows user to change password and submit to database.	SidebarPro

EventsSearch
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Enables the user to set different filters and search for events.	- Events - EventsDisplay

SearchContext
Parent/Subclasses: Context

Responsibilities:	Collaborators:
Stores the search filters as global states so they can be applied on the events page.	- EventsSearch - Events - EventsDisplay

Update Events
Parent/Subclasses: n/a

Responsibilities	Collaborators
- For a user to update an event to the database. - Allows user to specify title, description, location, image, time, date, category, and whether or not the event is intended for women only - When update is posted, an email is sent to all attending users.	- Side bar - After an event is added it will appear on the event page of all users. - Can be added to the liked events page by any user. - If a user attends this event it will appear in the history page

Add Notification
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- For a user to add a notification for a specific event - Allows user to specify title and description	- SidebarPro - Events - Notifications

UserNotifications
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Page that shows all user's notifications	- SidebarPro - Events

EventContext
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Share states between components on the single event page if necessary.	- Event - AddComment

AddComment
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
Text area / form for adding comments as a logged in user.	- Event - EventContext - EventPageSingular

EventComment
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
● Shows a single user comment on the singular event page.	● Event ● EventPageSingular

Backend CRC

Search	
Parent/Subclasses: n/a	
Responsibilities: ● Return an instance of every event in the database ● Return an instance of a user from the database given their id ● Return events through a filtered search with specified parameters (category, woman-only events, city, etc.)	Collaborators: ● MongoDB eventual Database

Add Event	
Parent/Subclasses: n/a	
Responsibilities: ● Add an event according to the specified parameters (name, description, image, etc.) ● Attach address information (city, country, region, longitude, latitude, etc.) to events before insertion, to allow for easy geographical lookup	Collaborators: ● MongoDB eventual Database

MongoDB eventual Database	
Parent/Subclasses: n/a	
Responsibilities: ● Provides collections of data containing all the information users interact with while using the application.	Collaborators: ● n/a

Event
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- Provides a model for events that users can create, like and sign up for. - Defines how the event will be formatted in the MongoDB database *eventual* - Provides the format that events are returned when called from the database to be displayed/interacted with in the frontend. - Sends emails when to attending users when event is updated or when notification is posted.	MongoDB eventual Database

User
Parent/Subclasses: n/a

Responsibilities:	Collaborators:
- Provides a model for users that contains their username, encrypted password, email, and name object with first and last name. - Provides the jwt token for authentication - Provide current user access - Defines CRUD operations for users.	MongoDB eventual Database

LikesDAO
Parent/Subclasses: n/a

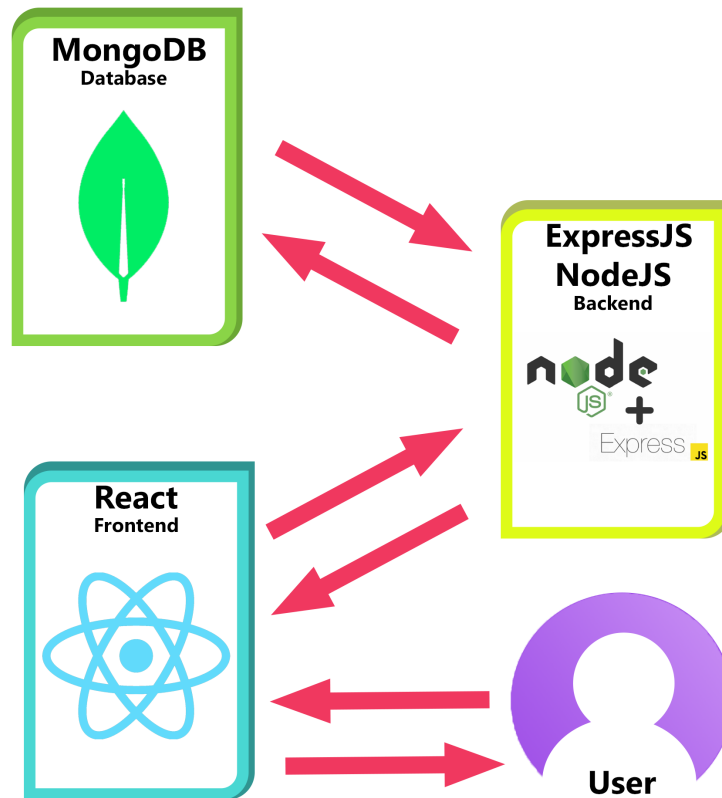| Responsibilities:	Collaborators:
- Provides functionality to create and delete a like and a join object from MongoDB - Provides functionality to update	MongoDB eventual Database

the num_likes and num_joined nufield in each object of testEvents collection in MongoDB.	

PasswordReset	
Parent/Subclasses: n/a	
Responsibilities: ● Provides functionality to create PasswordResetTokens, add them to a user, and send the token url to the user. ● Provides functionality to check if token is valid by checking database and expiry time. ● Provides functionality to update password and reset token.	Collaborators: ● MongoDB eventual Database ● Nodemailer ● Bcrypt ● Crypto

Notifications	
Parent/Subclasses: n/a	
Responsibilities: ● Provides functionality to create a new notification object for the database	Collaborators: ● MongoDB eventual Database

Comments	
Parent/Subclasses: n/a	
Responsibilities: ● Provides functionality to add a comment to an event.	Collaborators: ● MongoDB eventual Database

Software Architecture

We will be building our project on a 3-tier architecture using the MERN Stack. Having as a 1st tier React which is responsible of the front end web application, 2nd tier ExpressJS and NodeJS acting as a middleware responsible of being the runtime environment and server application framework and at 3rd tier MongoDB which is our No-SQL database.
Reference to the 3-tier architecture: https://www.mongodb.com/mern-stack