

RUNTIME WARRIORS

System Design Document

v.Sprint2

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

Table of Contents

Title Page	1
Table of Contents	2
Frontend CRC	3
Events Page	3
Event Page (singular)	3
Events Display	3
EventCard	3
Add Events	4
Liked	4
Login Page	4
Register	5
RegisterContext	5
SidebarPro	5
Profile	5
ForgotPassword	6
ResetPassword	6
EventsSearch	6
SearchContext	6
Update Events	7
Backend CRC	7
Search	7
Add Event	7
MongoDB eventual Database	8
Event	8
User	8
LikesDAO	8
PasswordReset	9
Software Architecture	10
Diagram	10

Frontend CRC

Events Page	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none">• 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	Collaborators: <ul style="list-style-type: none">• Search• EventsDisplay• Event Page (singular)

Event Page (singular)	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none">• Get an instance of a specific event from the backend• Display the information of said event including name, time, date, city, etc.	Collaborators: <ul style="list-style-type: none">• Event

EventsDisplay	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none">• Given any collection of events, display them all to the user in a compact view	Collaborators: <ul style="list-style-type: none">• EventCard

EventCard	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none">• Given an instance of a singular event, display it to the user in a compact view	Collaborators: <ul style="list-style-type: none">• n/a

Add Events	
Parent/Subclasses: n/a	
Responsibilities <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> Provides the logged in user a list with the events they liked 	Collaborators: <ul style="list-style-type: none"> SidebarPro LikesDAO

Login Page	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> If registered <ol style="list-style-type: none"> the page provides the user the ability to log into their account The user is able to access user-only features once logged in, such as join events, get notifications, etc If not registered <ol style="list-style-type: none"> The page is able to redirect to the register page, and the user can register Using JWT token, remember the logged in user 	Collaborators: <ul style="list-style-type: none"> SidebarPro User Register

Register	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • Updates global state when user clicks outside of Register. • Updates user info state (username, password, email, firstname, lastname) per keystroke. • Posts user info when submit button is clicked. 	Collaborators: <ul style="list-style-type: none"> • RegisterContext • User

RegisterContext	
Parent/Subclasses: Context	
Responsibilities: <ul style="list-style-type: none"> • Stores global state to be used in other classes. 	Collaborators: <ul style="list-style-type: none"> • n/a

SidebarPro	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • Act as a router between the different pages of the application. 	Collaborators: <ul style="list-style-type: none"> • Events Page • Liked • Login • Register

Profile	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • 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 	Collaborators: <ul style="list-style-type: none"> • SidebarPro • localStorage

database.	
-----------	--

ForgotPassword	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> Allows user to submit password reset request by submitting email. If email is valid, /forgot-password is called. 	Collaborators: <ul style="list-style-type: none"> SidebarPro

ResetPassword	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> 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. 	Collaborators: <ul style="list-style-type: none"> SidebarPro

EventsSearch	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> Enables the user to set different filters and search for events. 	Collaborators: <ul style="list-style-type: none"> Events EventsDisplay

SearchContext	
Parent/Subclasses: Context	
Responsibilities: <ul style="list-style-type: none"> Stores the search filters as global states so they can be applied on the events page. 	Collaborators: <ul style="list-style-type: none"> EventsSearch Events EventsDisplay

Update Events	
Parent/Subclasses: n/a	
Responsibilities <ul style="list-style-type: none"> • 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 	Collaborators <ul style="list-style-type: none"> • 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

Backend CRC

Search	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • MongoDB eventual Database

Add Event	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • MongoDB eventual Database

MongoDB eventual Database	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> Provides collections of data containing all the information users interact with while using the application. 	Collaborators: <ul style="list-style-type: none"> n/a

Event	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> 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 <i>eventual</i> Provides the format that events are returned when called from the database to be displayed/interacted with in the frontend. 	Collaborators: <ul style="list-style-type: none"> MongoDB eventual Database

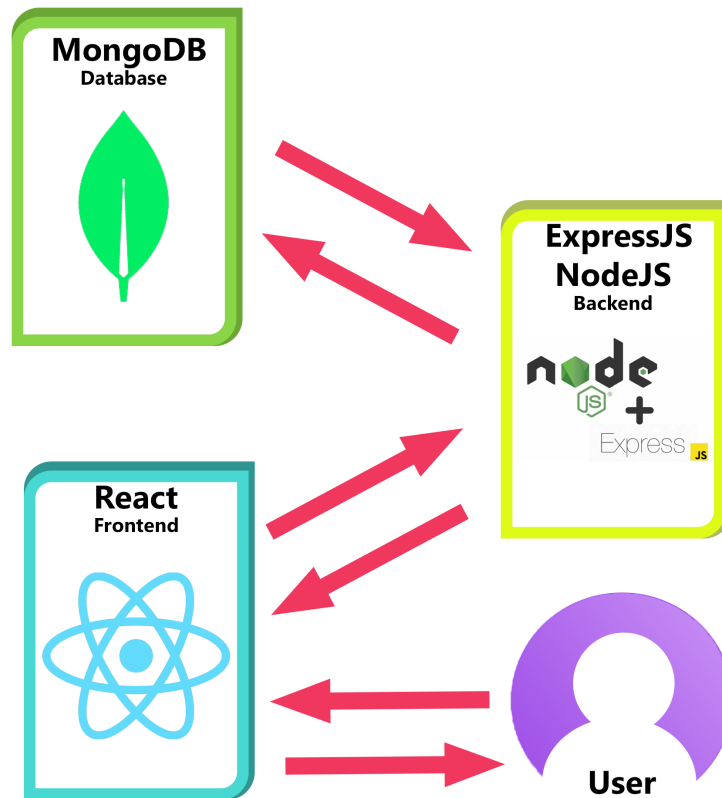
User	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> Provides a model for users that contains their username, encrypted password, email, and name object with first and last name. Defines CRUD operations for users. 	Collaborators: <ul style="list-style-type: none"> MongoDB eventual Database

LikesDAO	
Parent/Subclasses: n/a	

Responsibilities: <ul style="list-style-type: none"> • Provides functionality to create and delete a like and a join object from MongoDB • Provides functionality to update the num_likes and num_joined field in each object of testEvents collection in MongoDB. 	Collaborators: <ul style="list-style-type: none"> • MongoDB eventual Database
---	---

PasswordReset	
Parent/Subclasses: n/a	
Responsibilities: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • MongoDB eventual Database • Nodemailer • Bcrypt • Crypto

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>