American Heart Screening Management System

Design Document

Carlo Campanini ccampanini2018@my.fit.edu

Chris Newberry cnewberry2018@my.fit.edu

Jack Dewey jdewey2018@my.fit.edu

Noah Wilson wilsonn2018@my.fit.edu

Led By:

Drew Dunkelberger ddunkelberge2018@my.fit.edu

Sponsored By:

Dr. Eraldo Ribeiro eribeiro@fit.edu

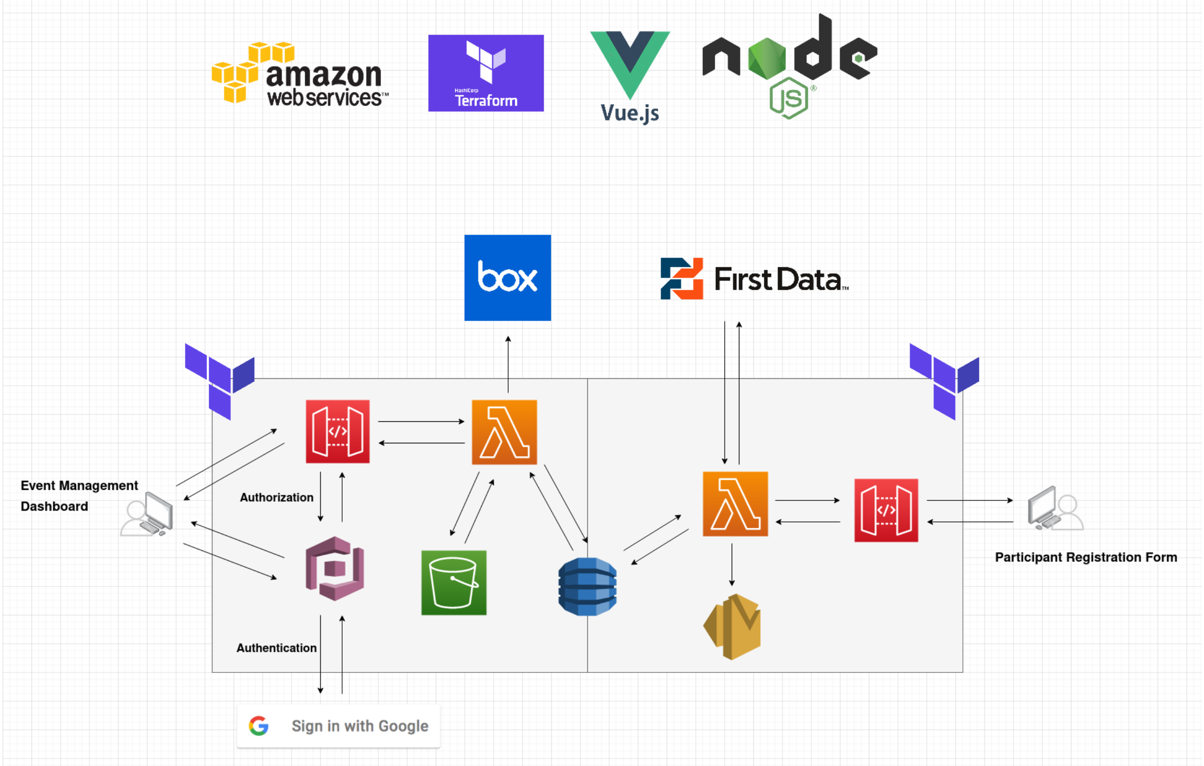
Client:

Evan Ernst, CEO - Who We Play For

Klynton Holmes, Tech Advisor - Who We Play For

Table of Contents

1. System Architecture Diagram



Description of main tools:

Vue.js : front end JavaScript framework used to build user interfaces.

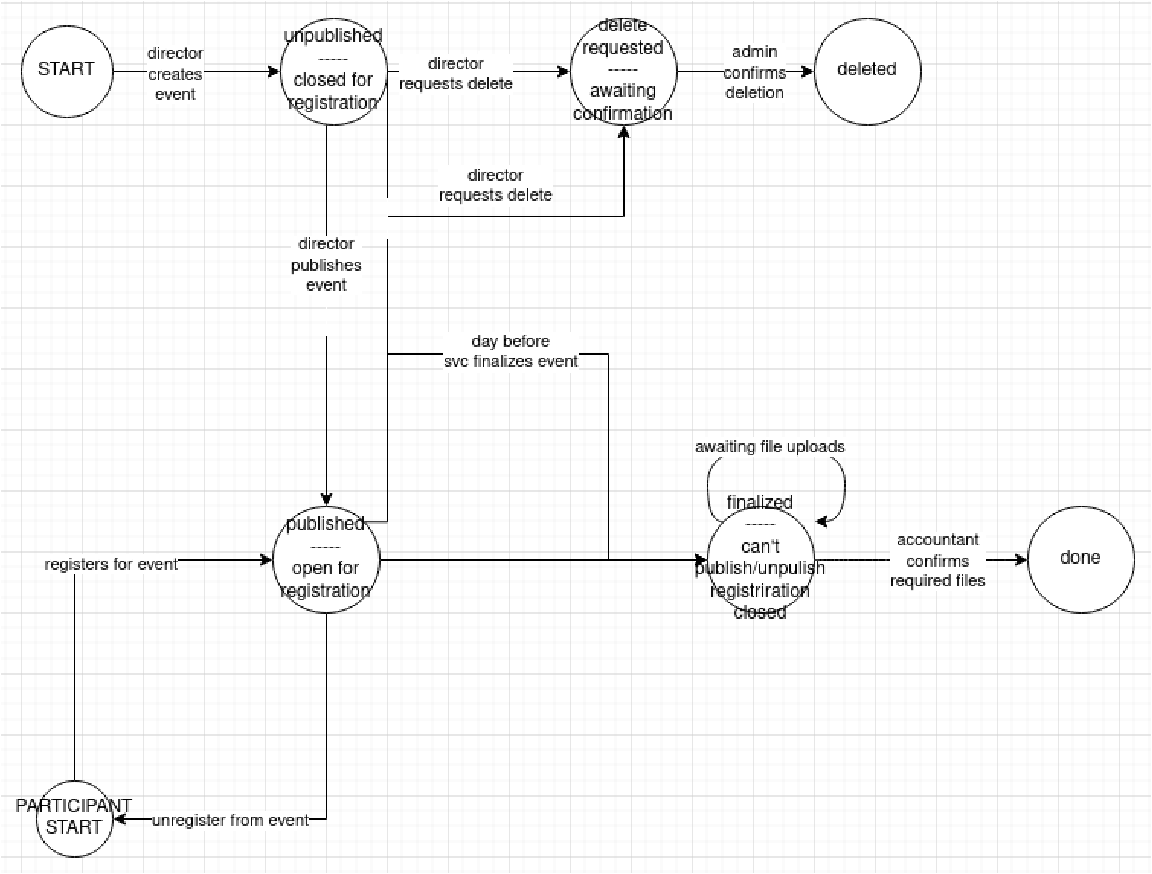
Node js : back end JavaScript environment that allows JavaScript code to run outside of a web browser.

Terraform : implements infrastructure as code to manage cloud services without the need for a console.

AWS Services:

- **DynamoDB**: NoSQL database that supports key–value and document data structures.
- **Cognito** : provides authentication, authorization, and user management.
- **API Gateway** : creates, publishes, maintains, monitors, and secures REST, HTTP, and WebSocket APIs at any scale.
- **S3** : object storage service that offers industry-leading scalability, data availability, security, and performance.
- **Simple Email Service** : (SES) email platform that provides an easy, cost-effective way to send and receive emails.
- **Lambda** : compute service that allows code to run without provisioning or managing servers.

2. Event State Diagram



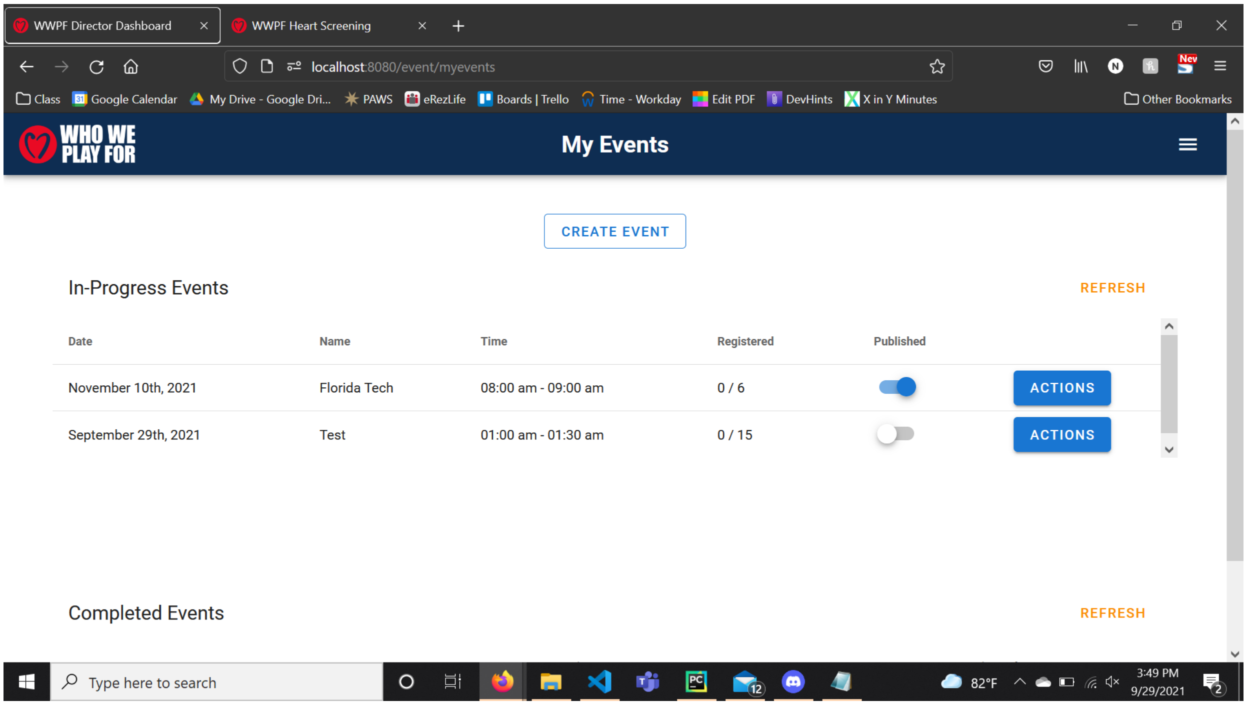
The event state diagram shown above displays all possible states an event can be in. An important note is that there are two start states, the "START" and the "PARTICIPANT START". The START state is the starting state for directors to create events, and the PARTICIPANT START state is the starting state for a participant, or user who is trying to register for an event.

Following the START state for directors, a director creates an event that is initially in an unpublished or closed for registration state. The director can then decide to publish the event so users can register for the event, or request deletion of the event. If the director requests to delete the event, it must await review from the administrator. Once the administrator handles any complications of event deletion (such as registered participants) and confirms, the event is deleted. If the director publishes the event, the event is now open for registration. Once the event is finalized, there is no longer an option to unpublish and registration for participants is closed.

Following the PARTICIPANT START state for participants, the participant has only one option of registering for an event. They can unregister and will end up back in the PARTICIPANT START state. Once they are registered, the system is in a state of waiting for the participant's file uploads. After the files have been uploaded and the accountant confirms them, the participant is finished.

3. User Interface

3.1. Director View



My Events

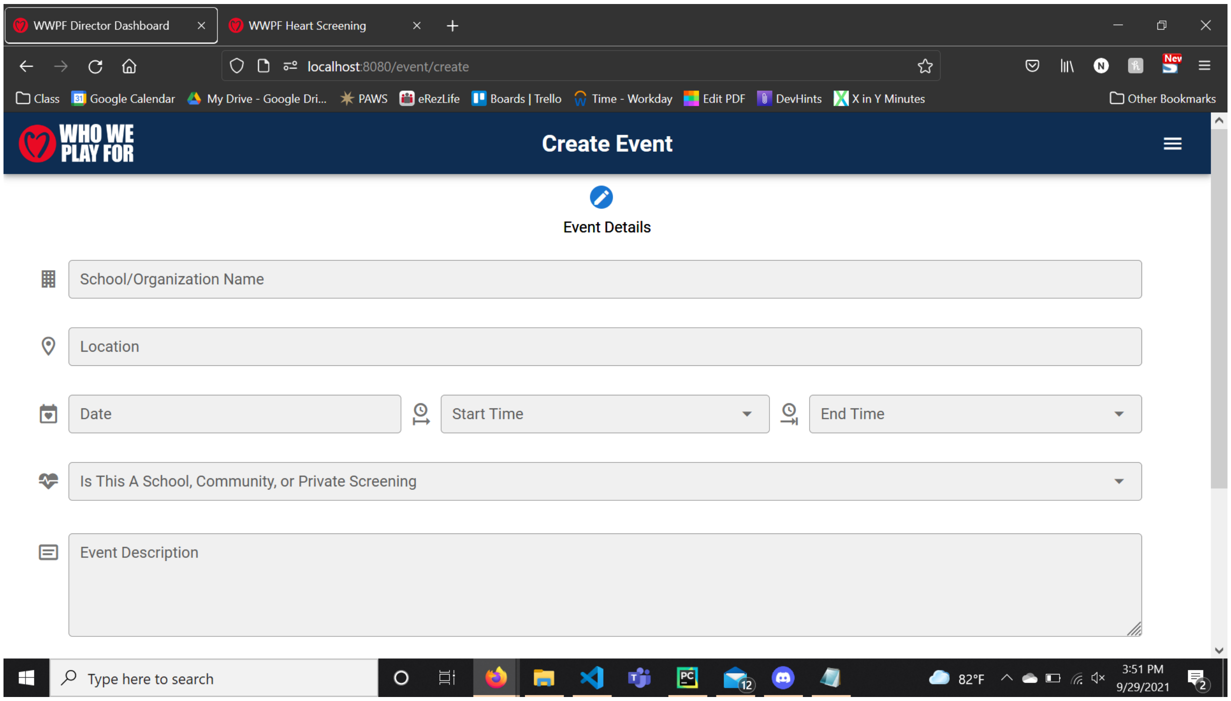
In-Progress Events

Date	Name	Time	Registered	Published	
November 10th, 2021	Florida Tech	08:00 am - 09:00 am	0 / 6	●	ACTIONS
September 29th, 2021	Test	01:00 am - 01:30 am	0 / 15	○	ACTIONS

Completed Events

This project is divided into two views, the director view and the participant view. When a director accesses the project, the screenshot above displays what they would see. The director view allows the directors to create events and then publish the events to be visible to participants.

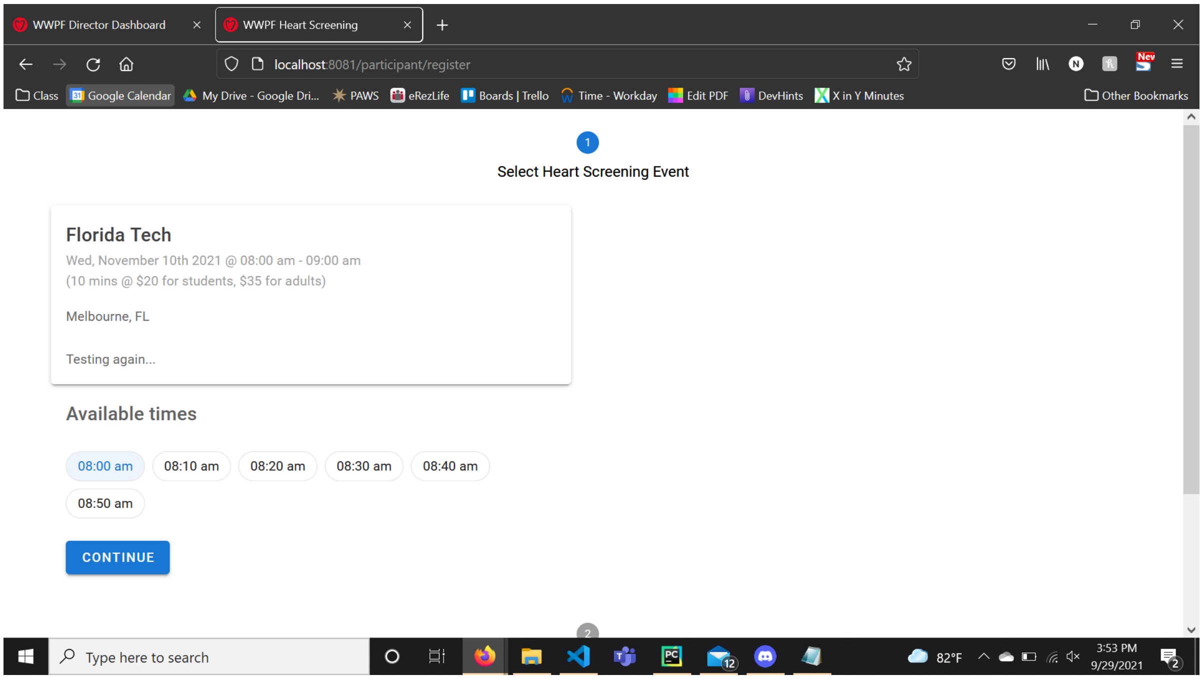
3.1.1. Creating an Event



After a director clicks the “Create Event” button, the screenshot above displays what they would see. This interface allows the director to describe, design, and put restrictions on the event.

3.2. Participant View

3.2.1. Selecting Event



Select Heart Screening Event

Florida Tech
Wed, November 10th 2021 @ 08:00 am - 09:00 am
(10 mins @ $20 for students, $35 for adults)

Melbourne, FL

Testing again...

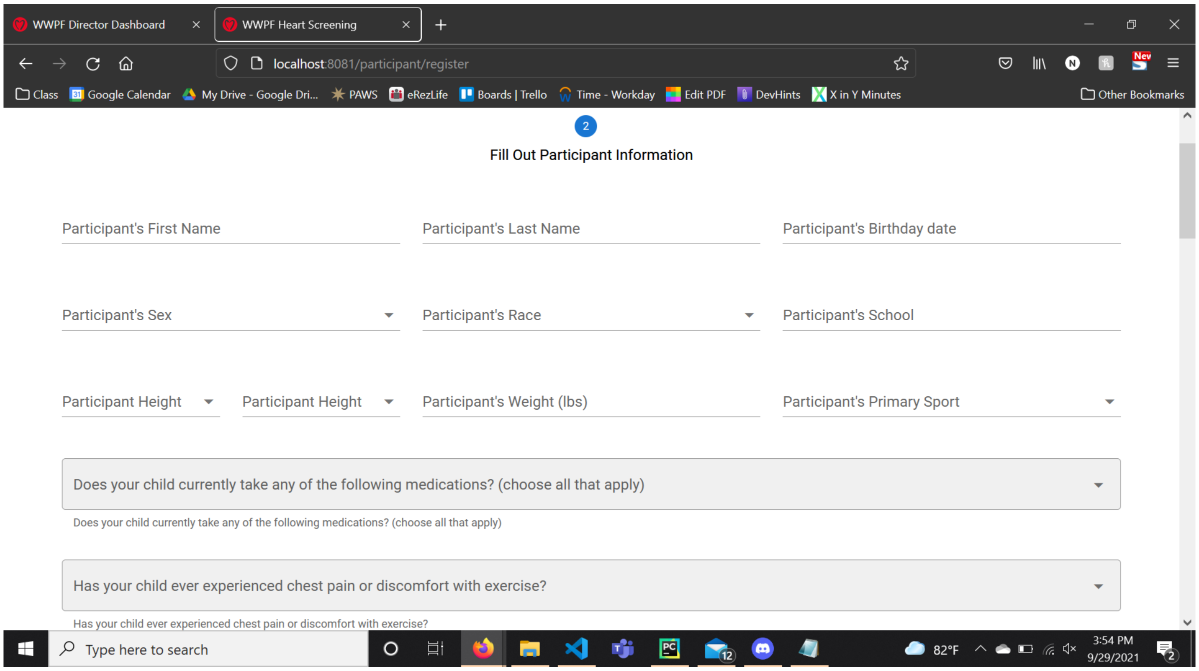
Available times

08:00 am 08:10 am 08:20 am 08:30 am 08:40 am
08:50 am

CONTINUE

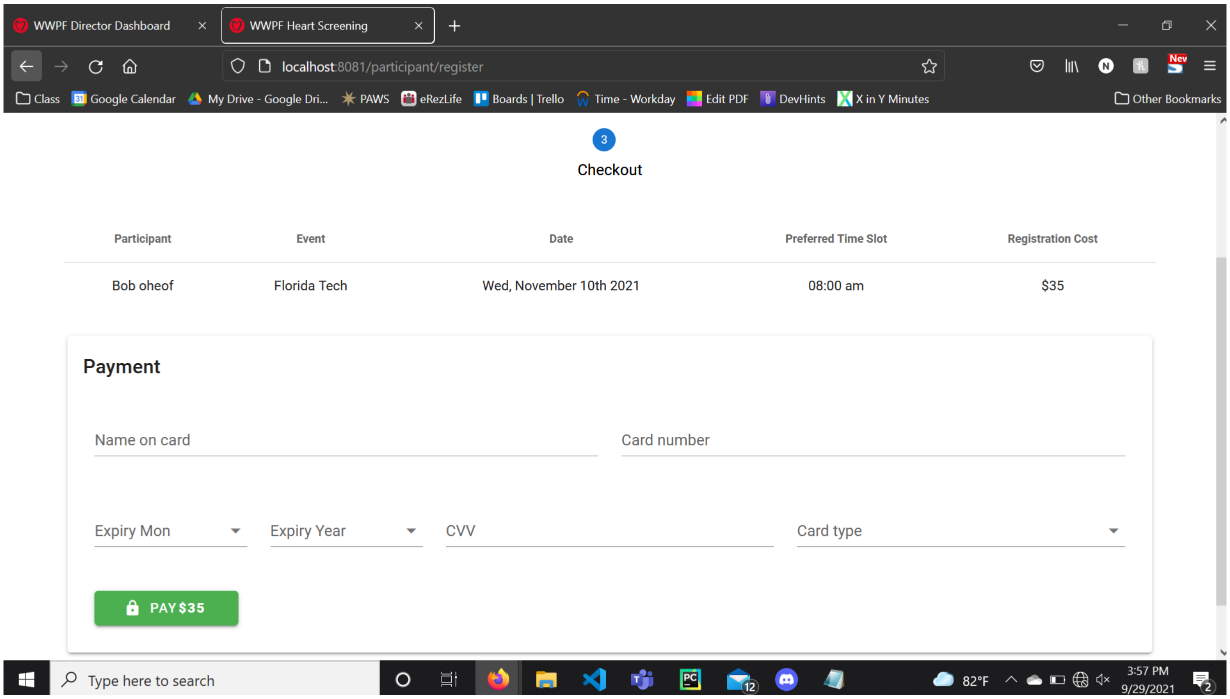
The screenshot above displays what a participant will see when they access the project. If there are multiple events setup, they will all be displayed in a list beside the event that is already on the screenshot above. After clicking on the event the participant wants to attend, the available times will be provided. Once the participant clicks a time, they can continue.

3.2.2. Filling Out Participant Registration


WWPF Director Dashboard
WWPF Heart Screening
localhost:8081/participant/register
Class
Google Calendar
My Drive - Google Dri...
PAWS
eRezLife
Boards | Trello
Time - Workday
Edit PDF
DevHints
X in Y Minutes
Other Bookmarks
2
Fill Out Participant Information
Participant's First Name
Participant's Last Name
Participant's Birthday date
Participant's Sex
Participant's Race
Participant's School
Participant Height
Participant Height
Participant's Weight (lbs)
Participant's Primary Sport
Does your child currently take any of the following medications? (choose all that apply)
Does your child currently take any of the following medications? (choose all that apply)
Has your child ever experienced chest pain or discomfort with exercise?
Has your child ever experienced chest pain or discomfort with exercise?
Type here to search
82°F
3:54 PM
9/29/2021

Once the participant selects an event and clicks the continue button, the screenshot above displays what they will see next. The participant will be asked to fill out information about themselves. Every field must be filled in order to continue.

3.2.3. Checking Out

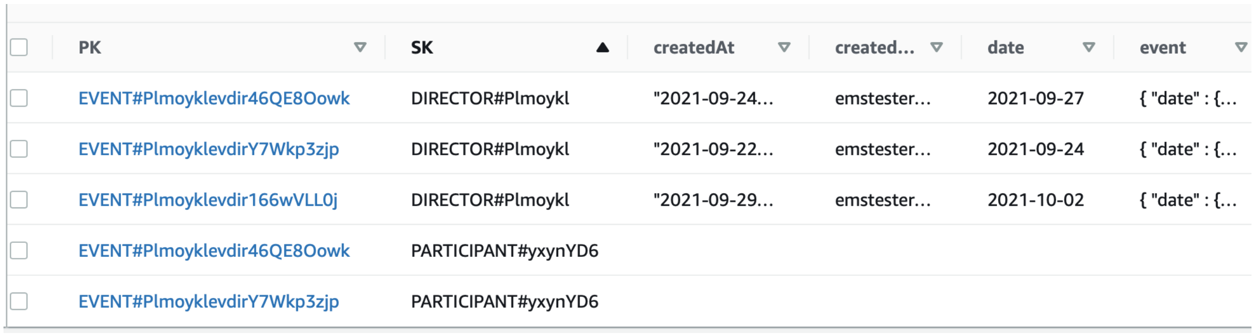


Participant	Event	Date	Preferred Time Slot	Registration Cost
Bob oheof	Florida Tech	Wed, November 10th 2021	08:00 am	$35

Once the participant has filled every field, the final view they will see is the checkout view. The participant will need to put their payment information in order to continue.

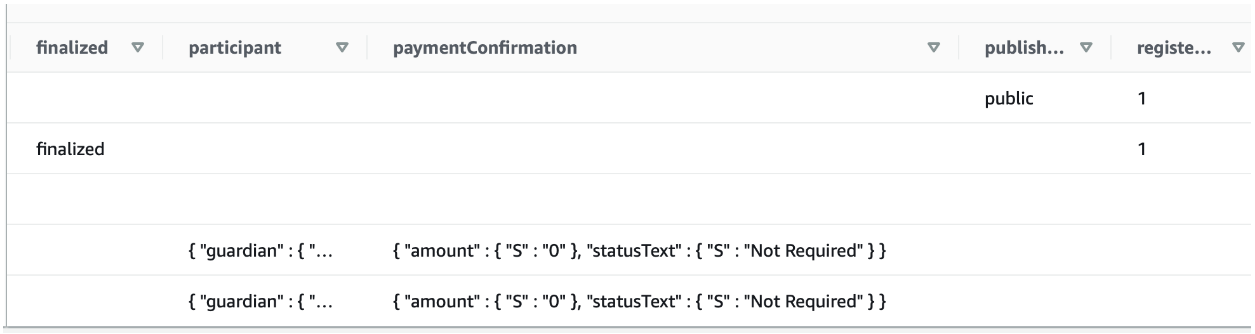
4. Database Design

This project uses **DynamoDB**, a NoSQL database. This allows for more flexibility as new features and requirements arise. In this table, primary keys are a combination of the partition key (PK) and sort key (SK). Looking at the tables below, we can see that events (DIRECTOR prefix in SK) and participants (PARTICIPANT prefix in SK) are stored in the same table. To associate a registered participant to a particular event, they are given the same PK.



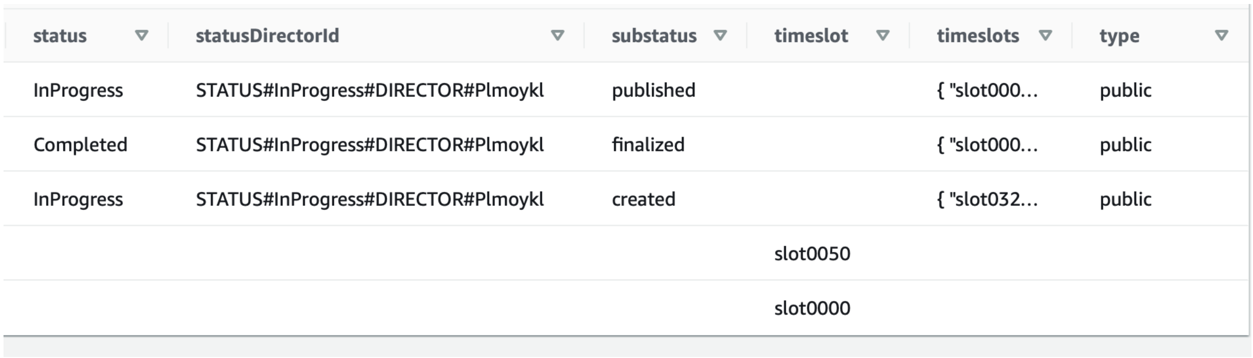
□	PK	SK	createdAt	created...	date	event
□	EVENT#Plmoyklevdir46QE8Oowk	DIRECTOR#Plmoykl	"2021-09-24...	emstester...	2021-09-27	{ "date" : {...
□	EVENT#PlmoyklevdirY7Wkp3zjp	DIRECTOR#Plmoykl	"2021-09-22...	emstester...	2021-09-24	{ "date" : {...
□	EVENT#Plmoyklevdir166wVLL0j	DIRECTOR#Plmoykl	"2021-09-29...	emstester...	2021-10-02	{ "date" : {...
□	EVENT#Plmoyklevdir46QE8Oowk	PARTICIPANT#yxynYD6				
□	EVENT#PlmoyklevdirY7Wkp3zjp	PARTICIPANT#yxynYD6				

Because of this, the table contains many several sparse indexes. Some of these, such as finalized, published, and statusDirectorId are also global secondary indexes, which can be used for efficient subqueries in place of the default PK and SK.



finalized	participant	paymentConfirmation	publish...	registe...
			public	1
finalized				1
	{ "guardian" : { "...	{ "amount" : { "S" : "0" }, "statusText" : { "S" : "Not Required" } }		
	{ "guardian" : { "...	{ "amount" : { "S" : "0" }, "statusText" : { "S" : "Not Required" } }		

Most of the data in the columns is stored as JSON objects, making it easy to access subcomponents of queried items in an object-oriented fashion.



status	statusDirectorId	substatus	timeslot	timeslots	type
InProgress	STATUS#InProgress#DIRECTOR#Plmoykl	published		{ "slot000...	public
Completed	STATUS#InProgress#DIRECTOR#Plmoykl	finalized		{ "slot000...	public
InProgress	STATUS#InProgress#DIRECTOR#Plmoykl	created		{ "slot032...	public
			slot0050		
			slot0000		

Project Structure

Libraries

Contains standalone libraries for authorization, logging, payment handling, and response handling which are necessary for the various sub-components of the project

Packages

- API: Contains backend source code and corresponding tests, separated based on role
- Core: Contains all of the business logic source code and corresponding tests, separated based on role
- DAO: Contains source code and tests for all interactions with the database containing event and participant information
- UI: Contains frontend source code, images, views, and plugins, separated based on role

Terraform

Contains infrastructure as code (IaC) for both the main components of the system (backend, frontend, and storage) as well as shared resources (authentication and certificates).