Heart Screening Management System

Team Members

- Carlo Campanini (ccampanini2018@my.fit.edu)
- Chris Newberry (cnewberry2018@my.fit.edu)
- Drew Dunkelberger (ddunkelberge2018@my.fit.edu)
- John Dewey (jdewey2018@my.fit.edu)
- Noah Wilson (wilsonn2018@my.fit.edu)

Faculty Sponsor

- Eraldo Ribeiro (eribeiro@fit.edu)

Client

- Evant Ernst, CEO - Who We Play For
- Klynton Holmes, Tech Advisor - Who We Play For

Meetings

Date	Discussion Points
August 25th, 2021	• Introduction - Chris and John • Summary of project and goals
August 30th, 2021	• Introduction - Full team • General planning • Description of tools needed
September 1st, 2021	• “Hello world” AWS demo • Milestone 1 planning
September 3rd, 2021	• “Hello world” TDD demo with Mocha and Chai • Demo of Terraform infrastructure tool and current architecture by client
September 7th, 2021	• Finalize milestone planning

Goal and Motivation

Motivation

Sudden cardiac arrest (SCA) is the 2nd most common cause of death in youth. It is also the leading cause of death for student athletes and death on school campuses. This is especially an issue for athletes with a heart condition that seem completely healthy on the surface, but are putting excessive strain on their heart. The common approach of a simple history and physical examination is not effective at detecting potentially fatal cardiac abnormalities in young adults. According to Dr. Joseph Marek, the standard physical examination misses over 96% of those at risk for SCA.

Goal

The ultimate goal of this project is to increase the use of ECG screenings in American high schools to minimize the cases of SCA in student athletes. This will be done by improving upon Who We Play For (WWPF)'s current heart screening program with an event-management system that provides automated processing of results, event scheduling tools, ECG collection, and results delivery. The current system is a pay-per-event, requires some setup, and is localized to the management of Florida users. The new system will be scalable for use across all of the United States.

Key Features

Feature	Description
On-site registration	System will support registration for participants on the site via QR code; It must be efficient, support a large amount of users in a short span of time, and be mobile-friendly
Scalable event scheduling and management	Directors will be able to create and manage events. Participants can register for events online or in-person. This will be done in a way such that the software

	is scalable to different regions/states
Automatic results delivery	System will provide automatic delivery of heart screening results to participants via text message and email once the ECG data is processed and interpreted

Novel Features / Functionalities

Feature	Description
Text notifications	Text notifications for reminders the day before and low-risk results delivery. The current system only sends emails for low-risk results delivery
Volunteer sign-up	The current system does not have a role for volunteers yet, making it difficult to manage them. The current roles are director, participants, accountants, and admins

Technical Challenges

Challenge	Description
Unfamiliarity with JavaScript and relevant frameworks (NodeJS and VueJS)	JavaScript is used for both the frontend and backend of the system, which is new to most of the team
Limited knowledge of Amazon Web Services (AWS)	The existing system makes use of AWS, but only 1 team member has experience with it (certification)
Handling sensitive information and money properly	We cannot make mistakes on payments since we are handling other people's money and personal information
Unfamiliarity with existing codebase	We are building upon an existing codebase and need to learn the current

	architecture, dependencies, and project structure
Getting familiar with specific testing tools (Mocha and Chai) and test driven development development (TDD)	The current software relies on Mocha and Chai as the test runner and assertion libraries. We will need to learn these tools as well as the TDD mentality for developing new features

Milestone 1 (Oct 4)

- Provide a demo using the selected AWS tools
    - S3 (VueJS) for object storage and UI
    - API Gateway for easy creation and management of APIs
    - Lambda (NodeJS) for serverless, event-driven communication between the frontend and backend
    - DynamoDB for scalable storage of data
- Provide demo for test tools, Mocha and Chai
- Resolve technical challenges
- Compare and select collaboration tools
    - Gitlab, Github, Slack, Discord, Google Drive
- Create a requirements and design document
- Create a plan for testing
- Create a payment management system
    - Allow finance person to search payments by:
        - Participant
        - Event
        - Date
        - Date range
    - Ability to export payments to spreadsheet
- Implement appointment rescheduling functionality

Milestone 2 (Nov 1)

- Implement, test, and demo online private event functionality and onsite registration
- Implement, test, and demo secure participant retrieval of ECGs

Milestone 3 (Nov 29)

- Implement, test, and demo a volunteer role
    - User interface and backend
    - Make it easy for volunteers to sign up for events

Task Matrix (Milestone 1)

Task	Carlo	Chris	Drew	John	Noah
Simple demo of architecture using AWS	Storing message to retrieve in DynamoDB and handling permissions	Develop Frontend using S3 (VueJS)	Handling request message via Lambda (NodeJS)	Handling request message via Lambda (NodeJS)	Develop Frontend using S3 (VueJS)
Simple demo of test framework	Experiment with Mocha and Chai individually	Experiment with Mocha and Chai individually	Experiment with Mocha and Chai individually	Experiment with Mocha and Chai individually	Experiment with Mocha and Chai individually
Resolve technical challenges	Gain more experience working with JavaScript as well as services like Terraform. Continue to add onto prior knowledge of AWS to help the team with any issues	Become more exposed to AWS and the services it offers to create web based applications. Develop a better understanding of database concepts	Gain experience with Terraform for AWS resource management, learn JS frameworks, and practice TDD	Develop stronger skills with javascript frameworks, testing tools, and AWS. Spend time looking at the current codebase	Become more exposed to AWS and the services it offers to create web based applications. Develop a better understanding of database concepts

Compare and select collaboration tool	Discord vs. Slack	Video conferencing tools	Github vs. Gitlab	Github vs. Gitlab	Video conferencing tools
Create requirements document	Write 10%	Write 35%	Write 35%	Write 10%	Write 10%
Create design document	Write 35%	Write 10%	Write 10%	Write 10%	Write 35%
Create test plan	Write 15%	Write 15%	Write 15%	Write 40%	Write 15%
Create a payment management system	Full-stack development and AWS resource management	Frontend development	Backend development and unit testing	Backend development and unit testing	Frontend development
Implement appointment rescheduling functionality	Full-stack development and AWS resource management	Frontend development	Backend development and unit testing	Backend development and unit testing	Frontend development

Approval

- "I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____