



Technology Director

HackPSU

A Comprehensive Guide

Authors:

Sushrut SHRINGARPUTALE

sush.shring@gmail.com

Matthew HEILMAN

mdh5389@gmail.com

Contents

I	Introduction	4
1	Checklist	6
2	Essentials	7
2.1	Task Descriptions	7
2.1.1	Pre-event	7
2.1.2	Day-of event	8
2.1.3	Post-event	9
2.2	Accounts	9
2.2.1	Google Cloud Platform (GCP)	9
2.2.2	Amazon Web Services (AWS)	11
2.2.3	Namecheap	11
2.2.4	Github	11
2.2.5	Alibaba	12
2.2.6	OneSignal	12
2.2.7	Asana	12
2.2.8	Travis CI	12
2.2.9	Devpost	13
2.3	Team Structure	14
2.4	Important Resources	14
2.4.1	Christy Long (Advisor)	14
2.4.2	Owned Hardware	14
2.4.3	Recommended Practices	14
II	Projects	16
3	Frontend	17
3.1	Overview	17
3.1.1	Landing page	17
3.1.2	Registration App / Live Page	17
3.2	Known bugs	18
3.3	Future features	18
4	Backend	19
4.1	Main Database	19
4.1.1	Overview	19
4.1.2	Known bugs	20
4.1.3	Future features	20

4.2	Redis Database	21
4.2.1	Overview	21
4.2.2	Known bugs	21
4.2.3	Future features	21
5	RFID Scanners	23
5.1	Overview	23
5.2	Known bugs	23
5.3	Future features	23
6	Admin App	24
6.1	Overview	24
6.2	Known bugs	24
6.3	Future features	24

Part I

INTRODUCTION

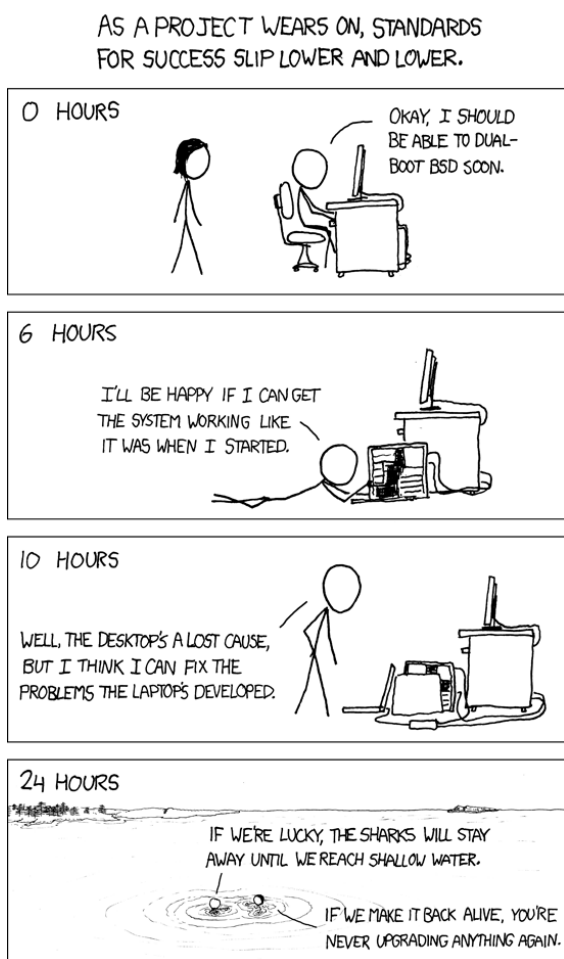
Congratulations on your acceptance as Director of Technology! We're sure you were selected from a very competitive pool of applicants as the best choice for the role.

The Director of Technology has grown to be a very important role within HackPSU. A large portion of the success of the event hinges on the success of the tech team achieving their goals. Hence we believe that it is very important that you be prepared for anything the event may throw at you.

This document will provide a high-level overview of the responsibilities of the Director of Technology for HackPSU. It lists the necessary information and tasks needed to make HackPSU a successful event as far as the Tech Team's contribution is concerned.

The next page lists a number of tasks that should help you keep a track of all the tasks needed to be completed before, during, and after the event. The rest of the document goes into considerable detail about the goals and responsibilities of the Tech Team and the Director.

Figure 1: Tech Projects



Source: xkcd.com/349 [1]

1 Checklist

This page is a quick TL;DR of suggested tasks that need to be taken care of during the semester. This list is in no way comprehensive or mandatory, but it should give you a good sense of how to structure the semester. The list is formatted as *[item - suggested time to complete]*

Pre-event

- ☐ Budget Proposal - 14 WBE ¹
- ☐ Finalize Landing Page Design - 12 WBE
- ☐ Open Registration - 10 WBE
- ☐ Order RFID Bands - 5 WBE
- ☐ Synchronize Sponsors on website - Continuous process
- ☐ Finalize plan for live website and tracking system - 3 WBE
- ☐ Synchronize dietary restriction data with Logistics - 3 WBE
- ☐ Generate day-of Slack - 1 WBE
- ☐ Send email to all registrants with pin number - 1 WBE
- ☐ Finalize and update schedule on the database - 1 WBE
- ☐ Finalize and update workshops on the website - 1 WBE
- ☐ Finalize and update prizes on the Website/Devpost - 1 WBE
- ☐ Update times for live countdown timer - 1 WBE

Week-of event

- ☐ Talk to PSU IT and boost building internet capacity - 2 WBE
- ☐ Redirect hackpsu.org to the live website - 3 DBE
- ☐ Upgrade server and database capacity - 1 DBE

Post-event

- ☐ Downgrade server and database - 1 WAE ²
- ☐ Change landing page - 1 WAE
- ☐ Team Debrief - 1 WAE
- ☐ Change active hackathon on database - 2 WAE

¹Weeks before event

²Weeks after event

2 Essentials

Just the essentials that will help you succeed in your position. This should be a good enough reference for any information you may need during your term as director.

2.1 Task Descriptions

2.1.1 Pre-event

- **Budget Proposal**

One of the important tasks at the start of the semester is to come up with the budget for the Technology team. This budget needs to be submitted to the Finance team at the very start of the semester. The budget does not need to be extremely accurate, but should reflect a good enough estimate of the expenses of the semester. This helps the Finance team keep track of expenditure by each team. For reference, refer to old hackathon budgets.

- **Finalize Website Design**

This does not mean that the whole design of the website be ready and coded up, but a general idea of what the website will end up looking like will help keep the Frontend project on track. Ideally, this task should be done in collaboration with the Design team.

- **Open Registration**

This task is fairly straightforward. Test out that registration works properly for the current hackathon and open it up to the public.

- **Order RFID Bands**

Order the RFID bands based on the current number of registrations. We recommend using manufacturers listed in section 2.2.5. In case a design change is needed, order the bands further in advance as they may take extra time to confirm the design with the vendor. Another thing to keep in mind is the Chinese New Year which forces all factories to shutdown for about two weeks. This occurs around February and affects the Spring hackathon.

- **Synchronize Sponsors on website**

Whenever sponsors are confirmed by the Sponsorship team, their logos should be added to the landing page. Depending on the tier, the logo size should vary.

- **Finalize plan for live website and tracking system**

Decide what data will be displayed on the live website and start asking other teams for the data you will need to have a functional live website. Test and stress-test the tracking (RFID Scanner) system to ensure smooth check-in. Some of the information needed includes workshop data, event schedule, prizes for challenges.

- **Synchronize dietary restriction data with Logistics**

Logistics requires that they are provided an estimate of how many special meals to order for the event. We collect information about dietary restrictions (Halal, Kosher,

Vegan, etc.) on the registration form, which should be shared with Logistics. The registration form should also note that any requests made after the 3 week mark will most likely not be fulfilled.

- **Generate day-of Slack**

Sponsors and participants like having a day-of Slack to get updates, information, and help during the event. A link to this workspace should be made available on the live website and included in the email send to participants a week before the event.

- **Send email to all registrants with pin number**

We send out an email to all registered participants containing their pin for registration. This email should also contain important event information like the schedule, workshops, and location. This email is an opportunity to remind hackers that they have registered for HackPSU and should come to the event. To send this email, craft up the HTML and use the Admin app's email send feature which generates emails unique to each hacker based on their information. Registrations completed after the initial email is sent out should receive this email after they register.

- **Finalize and update schedule on the database**

This is fairly straightforward as well. Use the admin app to update all the schedule information including event rooms, descriptions, and times in the database. Make sure all the information is correct from the Education and Logistics teams.

- **Finalize and update workshops on the website**

If the live website contains a section listing all the workshops separately from the event, it should be updated with location, time, and descriptions. Make sure all information is correct from the Education team.

- **Finalize and update prizes on the website**

A good way of listing out all the prizes for the hackathon is to link to Devpost and create the list of prizes there. This allows you to change the prizes whenever needed and also declare winners for them after the event easily.

- **Update times for live countdown timer**

The live website timer has hardcoded Unix times for when the hackathon starts and ends. The constants in the code should be updated to refer to the new times for the hackathon. Keep daylight savings in mind if it becomes a factor.

2.1.2 Day-of event

- **Upgrade server and database capacity**

Remove any replication caps from the server to allow it to scale as needed for check-in and the rest of the event. Upgrade the database to use a larger machine and we recommend also creating a failover replica (Easily doable through the Google Cloud Console).

- **Talk to PSU IT and boost building internet capacity**

PSU IT configures any special wireless networks we may need and also boosts the capacity of the PSU network in the building to be able to handle the nearly 1000 participants. Talk to Christy Long or IT directly to have this handled.

2.1.3 Post-event

- **Downgrade server and database**
Downgrade the server and database back to the original configuration to prevent unwarranted costs.
- **Change landing page**
The landing page should be changed to a thank you page after the hackathon in preparation for the next event.
- **Team Debrief**
We recommend going through a team debrief meeting in order to gauge what went well during the event and what should be worked on for next semester.
- **Change active hackathon on database**
Once the hackathon is over, a Tech Admin (Refer to permissions in section 4 should create a new hackathon entry and mark it as the next active hackathon. This step means that you have successfully executed everything necessary for the current hackathon. Congratulations!

2.2 Accounts

This section lists all the associated accounts that the Tech Team comes in contact with. The passwords for all the accesses to these accounts are available in encrypted form on Google Drive.

2.2.1 Google Cloud Platform (GCP)

99% of all server-side computing power is leased from GCP (with the exception of the local Redis server (see section 4.2). This account is perhaps the most important of all accounts. This includes the API servers, SQL Database, Email service, Realtime-Database service, Storage, and Authentication.

All Google related management is handled through the *hackpsudev@gmail.com* account. We recommend using this account only for super admin³ access. Google allows other users to be added as managers of projects with varying levels of access.

GCP access is two-fold: Firebase and GCP itself.

- **Firebase:**

³sudo

Firebase provides easy access to Identity and Access Management (IAM) (read authentication). The docs for Firebase can be found at firebase.google.com [2]. Additionally, we use Firebase to host our landing page and realtime database.

- **IAM**

For authentication, Firebase provides Single Sign On (SSO) [3] using secure OAuth protocol. Authentication is provided through simple email or via third parties such as Google, Facebook, and GitHub.⁴

- **Hosting**

Our domain is purchased through Namecheap, but the landing page is hosted through Firebase. Namecheap simply redirects requests to hackpsu.org to Firebase. The benefit of using Firebase is the ease of use and the free SSL certificate that they provide to any website hosted there.

- **Realtime Database service**

The Realtime Database is an interesting piece of technology that allows clients to subscribe to updates to the database. Any addition, change, or deletion from the database is pushed as an update to the clients subscribed to it. We use this to provide live updates during the event. The database is very scalable and extremely fast in pushing out updates. It also abstracts away all the difficulty in creating a "realtime messaging" system.

Managing Firebase systems are handled through the Command Line Interface (CLI). Download it at firebase.google.com/docs/cli

- **Google Cloud Platform:**

All server-side computation is performed using the full Google Cloud suite (console.cloud.google.com). The following products from Google Cloud are in use by HackPSU currently:

- **App Engine**

App Engine provides compute power as needed based on incoming demand. We use App Engine to run the API server (See section 4).

- **Cloud SQL**

The main database used by HackPSU is hosted on Cloud SQL (See section 4.1).

- **Storage service**

Any large files that need to be stored online are stored here. This includes resumes from registrations, reimbursement receipts, and anything else the server may need to upload.

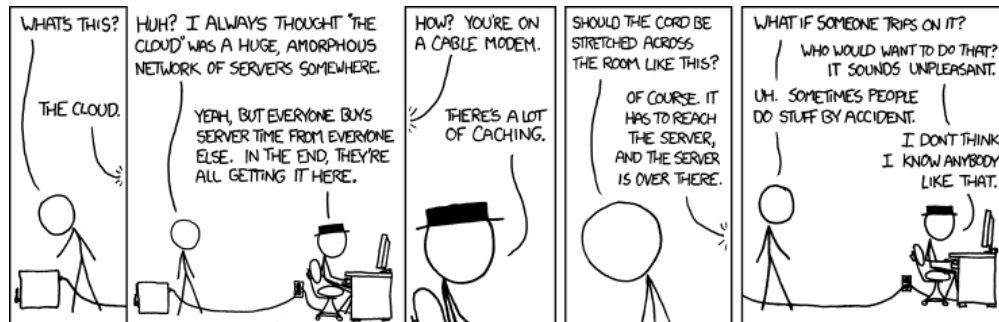
- **Logging and Monitoring**

While not exactly an active service used, StackDriver logging is one of the most important tools to ensuring that the server is functioning correctly. Debugging

⁴NOTE: We are not associated with Penn State authentication in any way. It is recommended that we inform hackers about this when registering to prevent them from using their Penn State email and password.

any issues on the backend should be directed here. This service keeps an eye of the responses that the server sends and aggregates any errors that may be occurring.

Figure 2: The Canonical Cloud



Source: xkcd.com/908 [1]

2.2.2 Amazon Web Services (AWS)

The only use of AWS currently is for hosting the built and compiled web applications (admin app [Section 6] and registration [Section 3]). This is necessary because Google Cloud Storage does not support hosting with SSL and Firebase can only host one website per project.

2.2.3 Namecheap

All domain management is performed at namecheap.com. This includes hosting records for the main landing page, sub-domain pages, service verifiers (mainly email service), and email forwarders. You will have to renew both hackpsu.org and hackpsu.com domains each year.

2.2.4 Github

All of HackPSU's source code is up on Github. All the repositories are owned by the Hack-PSU organization, which you should be an admin for. The organization page is at github.com/Hack-PSU. Most of the repositories on Github are integrated with Travis Continuous Integration (CI) (Section 2.2.8).

Some notes on the repo structure:

- The master branch of a repo should be protected and direct push to it should be prevented unless extremely necessary. This branch is where Travis should run its integration. Any code pushed to master should be production ready and depending on the CI configuration may be directly released to production. A separate development (`dev`) branch should be used to run all active development on.

- Build a habit of detailed information in commit messages. This helps track down changes and issues if they arise at any point.
- A healthy Git repository builds every new feature or change on a new branch, and each person works on a separate branch for that feature. Once the branch is good to go and ready to push into production code, it should be squashed into one commit to prevent the history of the production branch from becoming unreadable.
- A good idea is to make a release on all repositories for each event. This helps keep a record of what was used at each hackathon and can be used as a way to show changes between events.

2.2.5 Alibaba

Since we started using an RFID tracking system we have continually needed to acquire RFID wristbands. We have been doing so through Mrs. Jan Hu, a vendor on Alibaba. Some things to keep in mind when using Alibaba:

- These vendors are in China so there is a large time difference
- Vendors do not communicate well in English so be very specific in what you ask and in your responses.
- Save the invoice for reimbursement. If possible try to get something after paying for the order to make the reimbursement process easier.

2.2.6 OneSignal

OneSignal is a free service that provides push notifications on the live website for HackPSU. Push notifications are supported on Android devices and PCs currently. There are two projects on OneSignal currently configured - production and development. Any test push notifications should be sent on the development project, which can be configured to any GCS hosted application easily.

2.2.7 Asana

Asana is a project management tool we use for tracking progress within HackPSU. The Technology Team follows Agile principles while targeting goals, so Asana is an excellent tool to use for that. We highly recommend heavy use of the portfolio and timeline feature of Asana.

2.2.8 Travis CI

Travis is a useful [4] tool for automating the build, test, and deployment cycle for applications. Whenever a commit is pushed to master, our current configuration of Travis builds the code as defined by the build process, runs the tests defined in the code, and

deploys it to the appropriate source where it is hosted⁵. Travis is extremely simple [5] to learn and configure, but can perform very complex and powerful tasks in the automation process.

2.2.9 Devpost

Devpost (devpost.com) is an important piece of responsibility that the Tech Team is expected to handle. Each hackathon recognized by MLH is expected to have a new Devpost page that outlines the barebones of the hackathon, but more importantly, the prizes that are available for the event. Hackers expect to be able to submit their projects on Devpost, regardless of how the hackathon is performing project submissions and judging. Additionally, once the prizes have been declared, hackers expect that the projects that won be marked as such on Devpost. See help.devpost.com for more information on how to work with Devpost.

⁵Refer to the section on the individual projects below for specific hosting information.

2.3 Team Structure

The technology group is divided into four main teams: Frontend, Backend, RFID, and Admin App. The exact breakdown of each project is listed out in below sections. See Part II for details about each team's projects.

In the past, we have had a project manager that helps keep each project on track for the hackathon. Depending on the experience level of the members in the team, there may be a mentor (project leader) that helps newer members get familiar with the codebase. This person should also take charge of the project in question and work with the PM to keep it on track.

When selecting new members, it is important to take note of what skills the team needs to be successful, as well as judging the candidate by how much they are willing to learn. Learning is a crucial part of the HackPSU Tech experience, so this should be the cornerstone for any team member that is being added. Depending on what the demographics of the team are, we recommend looking for either fresher or more experienced students. This ensures that there is a good distribution of people that are willing to learn new things and take over in the coming semesters and team members that are excited to lead projects and have the technical skill-sets to make that happen.

2.4 Important Resources

2.4.1 Christy Long (Advisor)

Our current advisor is Christy Long. She is Director with Penn State IT, and a huge supporter of HackPSU. She has incredible contacts all across the board and in the past has helped us with mission crucial tasks such as implementing special WiFi at the event for hardware, coordination with Penn State IT for any other requirements, reaching out to companies (Microsoft, Apple, Google) for sponsorship, and any other administrative hurdles that we may have needed to cross while building up HackPSU. She is an amazing person and an incredible resource.

2.4.2 Owned Hardware

Presently, the Tech team owns the following:

- 1 - soldering iron
- 6 - RFID tracking system boxes (and one development version)

2.4.3 Recommended Practices

- First Github commit
When new team members are added, we recommend that they be asked to make a small change to a new branch. Their change should go through the full Pull-Request

procedure and then the branch should be deleted without merging. This gives them a good understanding of how the code review process works.

- **Attend one or more hackathons**

A very important experience for all team members is to attend one or more external hackathons. It provides team members with extremely important perspective about how other hackathons are run. It also acts as a means of generating ideas for new things we may want to implement for HackPSU.

Part II

PROJECTS

3 Frontend

The Frontend team is tasked with executing the design vision for HackPSU's online presence for the semester. This consists of two pieces - Landing page and Live website/registration application [6].

3.1 Overview

3.1.1 Landing page

The landing page is where hackpsu.org redirects to. It is build using plain HTML, SCSS, and JavaScript. The aim of the landing page is to provide the relevant information to anyone looking at HackPSU. The following information generally belongs on the landing page:

- Event Date and Location
- Link to registration
- Link to sponsorship packet
- FAQ and general event information
- Skeleton of schedule
- Sponsors
- Social media links

Hosting The landing page is hosted using Firebase. As Firebase provides free SSL certificates, this provides us the ability to access hackpsu.org using the https protocol and a modicum of security.

Since we use SCSS for the styling of the landing page, it must be compiled before it is uploaded to hosting.

3.1.2 Registration App / Live Page

We implemented a web-application for registration and the live page. This application is build using Angular [7] as it requires heavy interaction with backend and the user. Most features on this applications are protected by user authentication and are fairly customized to the user's information. As portions of this application have user-specific information, the following features were also implemented here.

- Travel reimbursement submission
- Project submission and table assignment

Content The live website contains information pertinent to the day of the event. The information on this page is much more dynamic and loads data from the live update source, live schedule, and a timer that counts down to the end of the hackathon. The following information is generally presented on the live website:

- Countdown Timer
- Link to Slack registration
- Live Updates
- Live Schedule
- List of workshops
- Other Day-of Information

Hosting The live website application is built using Angular and Webpack, which generates statically loadable, minified files. These files are hosted on AWS's CloudFront and [8] S3 as they have support for SSL certificates on any number of hosts. Currently, there do not seem to be any issues with the CI for this, meaning all pushes to `master` are directly built and uploaded to S3.

3.2 Known bugs

Travis Current Firebase admin authentication has some issues where the authentication token for CI environments expires very quickly. As a result, the continuous integration cycle for the landing page is broken and requires manual deployment.

Form Sanitization The travel reimbursement form, table submission form, and to a much lesser extent the registration form face input sanitization issues, especially across different platforms. This may cause some submissions to get rejected by the backend. The backend returns a well structures message in the JSON response, but it may not always bubble all the way to the UI properly.

3.3 Future features

Profile Page A much needed feature for the registration application is a profile page that allows the user to view and edit information about their registration. Currently the only way to do so is by emailing the HackPSU team directly and the changes being made directly to the database.

Extra Credit Registration Another useful to have feature would be providing hackers the ability to register for extra credit directly on the profile page. This would reduce a step during check-in and make it easier to manage.

Medium Integration on Landing page A proof of concept for this feature already exists as a pull request on the repo. This feature request was made by the Communications team. They would like us to feature recent posts from the HackPSU Medium account on the landing page in order to drive traffic to the blog.

4 Backend

The backend for HackPSU drives all the data and interfaces across all applications. It is written in JavaScript and is currently being migrated to TypeScript for better type-safety. The server runs on Node version 8+. A secondary server built on top of Redis is used locally during the event, and acts as a write-through cache for the hardware deployed in the venue. This redundancy is created to ensure that in case of network failures, the scanners continue to function with no issues.

4.1 Main Database

As explained above, the "Main Database Server" runs on Node v8+ and is deployed on Google Cloud's App Engine platform. The database backing the server is MySQL 5.6 which is deployed on Google Cloud's SQL service. The server is built as a RESTful service to provide sanitized access to the SQL Database.

4.1.1 Overview

The server processes HTTP requests from clients and provides them with data as requested. The server handles many parts of the various applications, and so it is required that it provide strong authentication, authorization, and encryption guarantees. To fulfill these, all requests to and from the server are encrypted using SSL over HTTPS. Any data access that is user specific requires the request to be made from an authenticated client with the proper permissions. The permissions model is explained below. Authentication and authorization support is provided by Firebase's Auth API, which integrates very well with all the front facing applications.

Permissions Model Permissions are broken into five categories as listed below:

- **No permissions/Participant** - Participants have the least permissions. They are allowed to submit any forms found on the registration app, read live updates, and read schedule information. They have the permission to change their information, but currently there is no support for that on the frontend.
- **Volunteer** - The next level of permission is a Volunteer. Volunteers are people that are helping organize the event day-of, but are not part of the team during the rest of the semester. They have access to the admin-app (Section 6) but cannot make any changes to any data.
- **Organizer** - Organizers are part of the team and have access to most features that will not completely affect the integrity of the data for the event.
- **Director** - Directors have almost full root access to make any changes to the data barring some database tables on whom the success of the event depends.

- **Tech Member** - Literally everything. Members of the tech team have full access to do anything they want to any data, anywhere. This permission level is the `sudo` of the HackPSU system, so it should not be handed lightly.

4.1.2 Known bugs

Type Confusion On some occasions, JavaScript's poor typing may face type issues with incoming data. However, in the current release of the server, these issues have been at an all time low. As the server continues to grow, these issues may grow again, which is why the migration to TypeScript has been suggested.

4.1.3 Future features

Analytics Engine Since there is an abundance of data available now, an interesting feature to implement would be an analytics engine that takes current hackathon data and provides insights for the teams to make informed decisions.

4.2 Redis Database

Part of the infrastructure for HackPSU is the Redis Server which helps IOT devices (scanners) connected to our network to maintain low latency between the scanners and our backend server. It also provides redundancy for when the network cannot send messages to the server because of the 800+ users on our network during events. Having a caching system setup will protect the system from losing messages that need to occur and allow our scanners to continue servicing users well beyond massive failure of our backend system.

Redis Server

- Runs Node v10+
- Runs on Docker or NPM/Node v10
- Runs Redis to host the in-memory cache
- Runs MongoDB to host the Permissions for the different User types.
- Uses Express.js to connect to the main server using a RESTful API

4.2.1 Overview

The server processes HTTPS requests from scanner clients and forwards the data that they provide as a write-back cache. This is ok because all scanners will access the cache before the server and won't need to confirm data written to the main server.

The server also provides Scanners with information through HTTPS using read-through cache. It allows the scanners to be updated with the latest user and location information.

The server handles many parts of the various applications, and so it is required that it provide strong authentication, authorization, and encryption guarantees. To fulfill these, all requests to and from the server are encrypted using SSL over HTTPS. Any data access that is user specific requires the request to be made from an authenticated client with the proper permissions. Since the server is on-premises, we had to self-sign the SSL certificates. To verify the authenticity of the server, we have hardcoded the fingerprint of the SSL certificate into each scanner to help it identify valid certificates.

Permissions

- Scanner API Key: Give's scanners access to the Redis server through HTTPS. It uniquely identifies each scanner and allows us to track which device is accessing what functions.
- Admin: Complete access to all features.

4.2.2 Known bugs

4.2.3 Future features

- Setup redis server to handle 1000+ requests per second

- Have smart caching procedures implemented
- Implement API to allow scanners to run future features from Admin app and Main Server API

5 RFID Scanners

5.1 Overview

This project is responsible for developing and maintaining each of the RFID scanners we are using to track attendance data. Currently, there are 6 functional enclosures. The hardware used is a NodeMCU, a RC522 RFID scanner, a 16x2 LCD display with an I²C driver, and a 4x4 keypad. Each of these components are soldered directly to a 3cmx5cm PCB board.

5.2 Known bugs

Hardware usage After much use, some of the boxes have developed poor connections at the micro USB port. Additionally, some of the LCD displays have been flickering due to use and had to be replaced.

NodeMCU The current configuration leaves no free pins for future features. Additionally, there were many development challenges slowing progress on this project.

5.3 Future features

Upgrading Circuitry The human aspect required to solder the connections was an unpleasant and imperfect process. Creating a custom PCB for each enclosure will allow highly precise and consistent results. Adding this will also help other events who which to adopt our system.

3D Printed Enclosure Having a 3D printed design will help the recreation of a box be quicker and less difficult than manually assembling a box. Additionally, not everybody has access to the tools required to make a wooden enclosure.

Sponsor Tracking System The thought behind this new system is to allow sponsors to keep track of who came to talk to them. They can either use this as a means to track the people they are interested in collecting their resume or to get a sense of how many people talked to them. Additionally, this should prevent people not attending the event from talking with sponsors until after they have checked in. This should be considered if the circuitry is updated.

Additional Features Some smaller features that would be beneficial include item check-in/check-out. Allowing a wristband to be re-associated might also be helpful if someone loses the wristband they received from check-in.

6 Admin App

The Admin App is an internal tool that allows team members and directors to view and manage various parts of the current hackathon. It is currently live at admin.hackpsu.org.

6.1 Overview

The Admin App is written with the Angular framework in TypeScript. It is only accessible to team members and directors within HackPSU who need to be manually given access to the application. Authentication is again provided by Firebase.

Hosting Since the application is built using Angular, hosting it is possible using static, built files on Cloudfront and S3.

6.2 Known bugs

Travis Due to some issue with how the Github repository is setup, there is an issue with configuring continuous integration with this repository. As a result, a push to master needs to be manually built and uploaded to Cloudfront and S3 in order to create a new release.

UI Cleanup There are many issues with the UI in this application. There are bugs with search features, display errors, and other UI/UX fixes that need to be handled for this.

6.3 Future features

Statistics and Visualization While there exists some implementation of data visualization on the app, it would be useful to expand this to generate more insights based on past and current data.

Financial Tool A highly requested feature is a financial tool for the Finance director to handle internal team finances and reimbursement requests. The vision is that any internal spending can be tracked through the admin app directly. Team members can submit reimbursement requests and receipts to the app and the Finance director can handle them on their own time and terms.

References

- [1] R. Munroe, “Success,” XKCD. [Online]. Available: <https://xkcd.com/349/> 5, 11
- [2] “Documentation — firebase.” [Online]. Available: <https://firebase.google.com/docs/> 10
- [3] C. Esplin, “Firebase authentication for web – how to firebase,” Dec 2016. [Online]. Available: <https://howtofirebase.com/firebase-authentication-for-web-d58aad62cf6d> 10
- [4] K. Stamas, “Continuous integration with travis ci,” Dec 2017. [Online]. Available: <https://medium.com/@katestamas/continuous-integration-with-travis-ci-fbf3b8aa3947> 12
- [5] dwyl, “Learn travis.” [Online]. Available: <https://github.com/dwyl/learn-travis> 13
- [6] H. Tech, “Hackpsu frontend,” <https://github.com/Hack-PSU/hackPSU-Frontend>, 2018. 17
- [7] “Angular docs.” [Online]. Available: <https://angular.io/> 17
- [8] “Use cloudfront to serve a static website.” [Online]. Available: <https://aws.amazon.com/premiumsupport/knowledge-center/cloudfront-serve-static-website/> 18