Database Application

Final Project Report

Author: Jimmy Nguyen
SUNY Oswego
June 14, 2017
CSC 490 Final Project

Abstract: This document describes the final project processes/steps for CSC 490 Business Information Systems at SUNY Oswego.

1. Introduction of Project Idea

The idea of the project is that a spectator can submit challenges to a user. A user would then have to perform the challenge or face a penalty (for every challenge not performed, donate $5 to X charity). Spectators would be able to see which challenges have been submitted and which have been completed. This would entertain the concept of audience interactivity when developing such a platform that can handle and submit requests. There will be a web portal which will match the PI platform built on JFX using OpenJFX on the latest Raspbian.

There are still things that have not been completely polished. Challenges could of course be approved first before they are submitted to the available challenges list (they are currently approved on web platform on submission). There are no places to put a charity url if you refuse the challenge as well as no option to refuse a challenge (it will float in your list forever). With the time allocated, I think that the barebones structure of this application of a database is sufficient enough to provide the main idea and not hinder conceptual components of the project.

2. Embedded/Web Database Platform Overview

This final project is implemented using a database and effectively distribute the core driver software through web and system application. There were 3 phases of implementation as well as their division of distributed developments.

- Finding a reliable database host and type (MySQL used on *FreeSqlDatabase*)
    a. Design database via ERD
    b. Normalize database
    c. Create database
    d. Test functionality

- Create a device platform on an embedded device (RPI 3B used)
    a. Plan out necessary hardware (board, case, gpio, networking)
    b. Purchase and assemble components
    c. Configure board and OS (software)
    d. Test functionality (OpenJDK/Python)

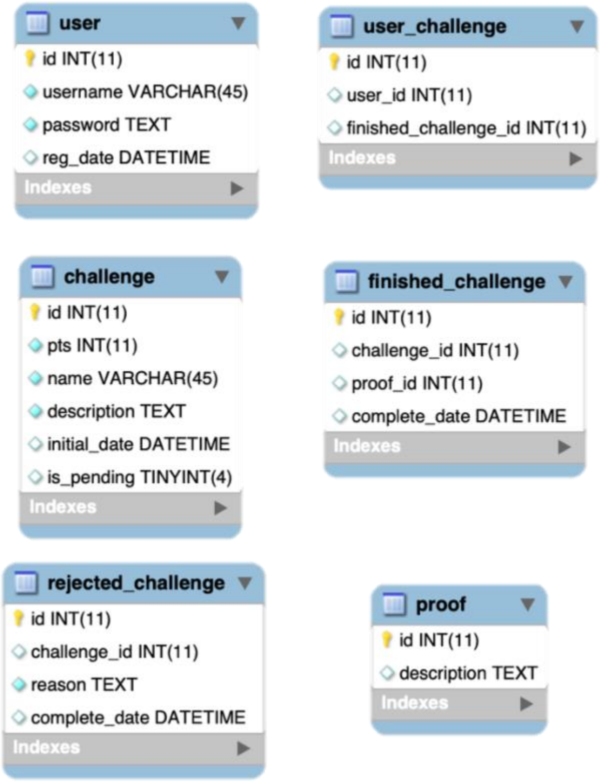
- Develop a software platform to interface with database (Java 8)
    a. Model platform environment
    b. Interface with database/network connectivity
    c. Program functionality & perform testing
    d. Design a graphical user interface
    e. Configure software with embedded device ctrls
    f. Deploy on embedded device for testing

- Develop a web-based platform to interface with database (PHP)
    a. Design platform via HTML/CSS
    b. Program backend with little to no resource leaks
    c. Perform security and functionality testing
    d. Deploy on pi.cs.oswego server as main web-host

3. Database Configuration

The database I used was MySQL 8.0.14 on the *FreeSqlDatabase* platform. I was given pi.cs.oswego MySQL credentials from Professor Odendahl but the connection was too slow and was causing > 3 second delays between screens as the software I wrote for the pi-platform is online persistent thus requires network and database connectivity through every window transition. Here is a diagram of our tables we created in MySQL.

Normalization was tricky but I was able to aggregate data via foreign keys (FK). They take the primary key of another table and link tables within one another. This is very different in terms of object oriented programming (OOP). In OOP you would have an object with a list of another object. In structured databases such as MySQL, we have tables that contain two FK's of two other tables and that is how we have a list of two tables (for example, pairing a user and finished_challenges table for user_challenges table which shows all the finished challenge by a user). Functionality was tested with many cases such as duplicity and passed so the next step was connecting our web and java application to the database.



user
- id INT(11)
- username VARCHAR(45)
- password TEXT
- reg_date DATETIME
- Indexes

user_challenge
- id INT(11)
- user_id INT(11)
- finished_challenge_id INT(11)
- Indexes

challenge
- id INT(11)
- pts INT(11)
- name VARCHAR(45)
- description TEXT
- initial_date DATETIME
- is_pending TINYINT(4)
- Indexes

finished_challenge
- id INT(11)
- challenge_id INT(11)
- proof_id INT(11)
- complete_date DATETIME
- Indexes

rejected_challenge
- id INT(11)
- challenge_id INT(11)
- reason TEXT
- complete_date DATETIME
- Indexes

proof
- id INT(11)
- description TEXT
- Indexes

4. Embedded Device Implementation

The project required a driver board. Many boards were available and put into consideration such as the ASUS Tinker, Rock64, PocketBeagle, and other variants of the Raspberry Pi (Zero W) but the RPI 3B was ultimate chosen for its form factor, available modules/cases, and availability of software compatibility. Below is a compiled shopping list from Amazon.

Name of Product	Description	Price
Raspberry Pi 3B (Element 13)	Main driver board	$35
Wingoneer Aluminum Enclosure Case with 2.2 TFT	Aluminum milled case with TFT and GPIO buttons to interface with our application.	$41.49
Adafruit Flex Cable 300mm/12”	Camera cable	$5.25
15 cm right angle Micro USB cable	For PC connectivity	$8.55
Velcro Strips 3.5” x .75”	For adhesion to power supply	$2.78
Attom Tech 2500 mAh Power Bank	Power bank	$14.95
		Total Cost: $108.02

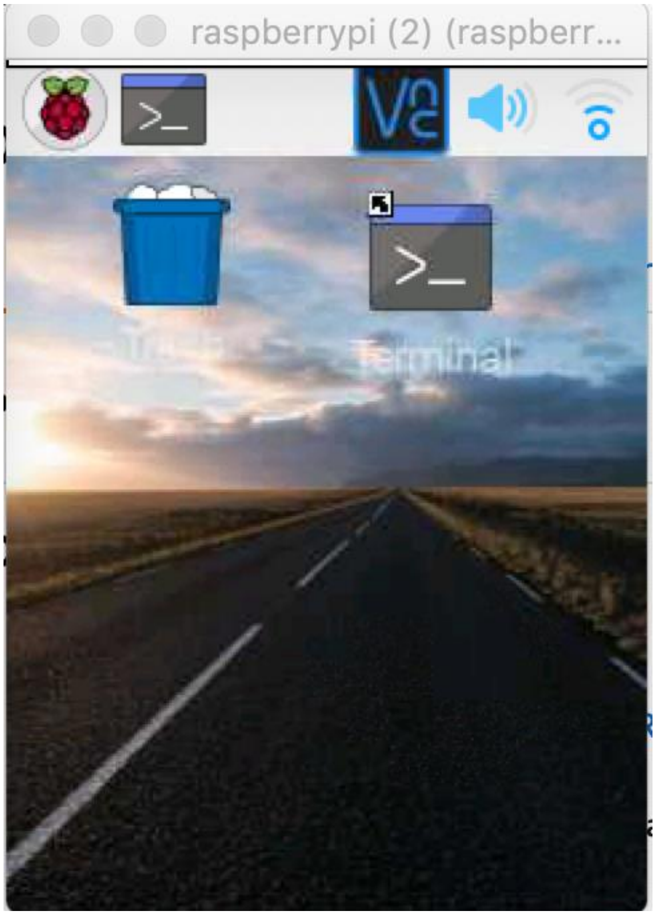
There was a slight problem with the Camera implementation. On challenge completion, you are supposed to be able to submit a photo of your challenge task but my camera module must have been defective as I do not get a readable connectivity signal. The camera idea was ditched after purchasing another flex cable and realizing the camera module was the problem.

The following were used for configuring the OS (Raspbian) environment

Configuration	Description
OpenJDK 8 **OpenJFK**	Runs Java 8 along with JavaFX
PIP **Python**	Interface software driver for GPIO connectivity
Modprobe **UINPUT**	For GPIO to keystroke driver
RealVNC	For remote connectivity and file transfer.

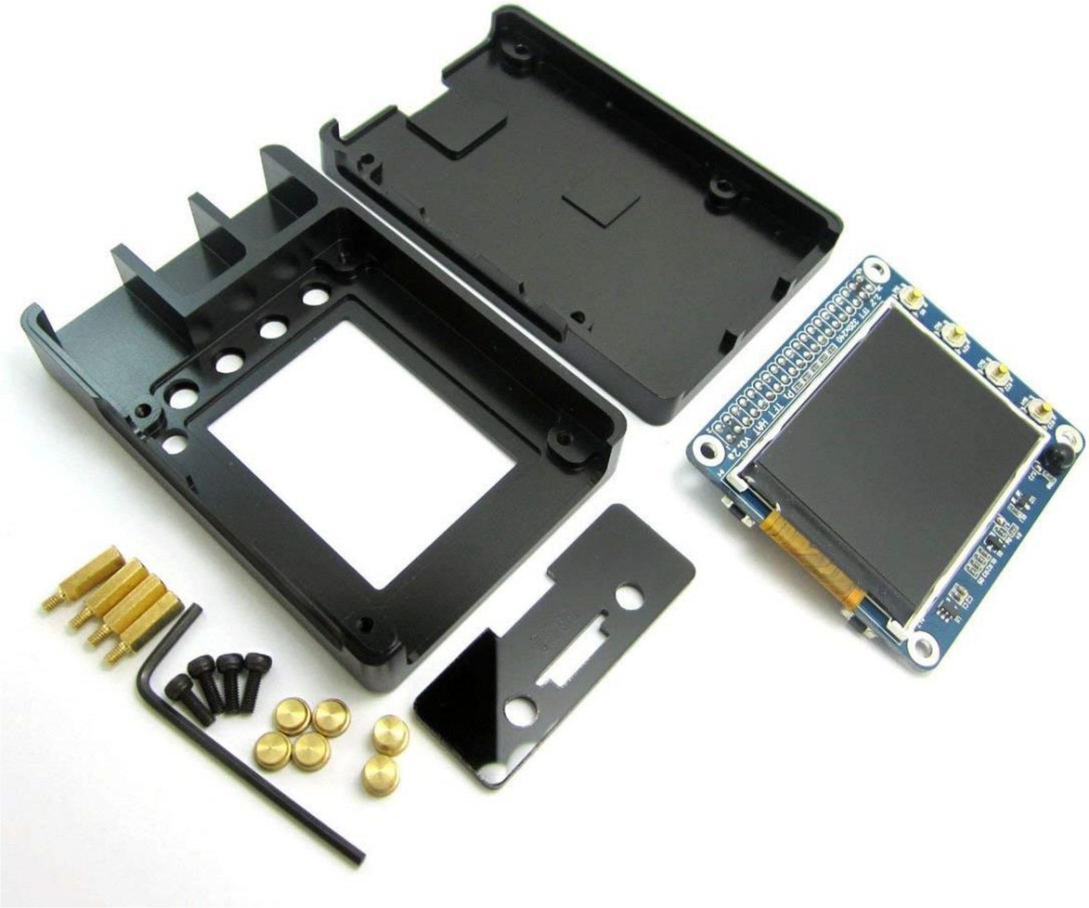
The drivers for the 2.2 TFT was downloaded and installed from:

http://www.raspberrypiwiki.com/index.php/2.2_LCD



Desktop of machine when VNC-ed into.

We run a sudo raspi-config to make sure our interfacing configuration is proper (camera, gpio, network). An error we had was with network connectivity. SUNY Oswego on the fourth floor does not allow guest users to register themselves (only by staff) and the enterprise network is not compatible with the RPI 3B's network card. We bypass this by hosting a 4g hotspot (iPhone 7). This also became a problem since the aluminum case interferes with our network signal so we have a weaker connectivity than we normally would with a barebone Raspberry Pi. A fix for this can be the use of a better networking card (perhaps USB network adapter).

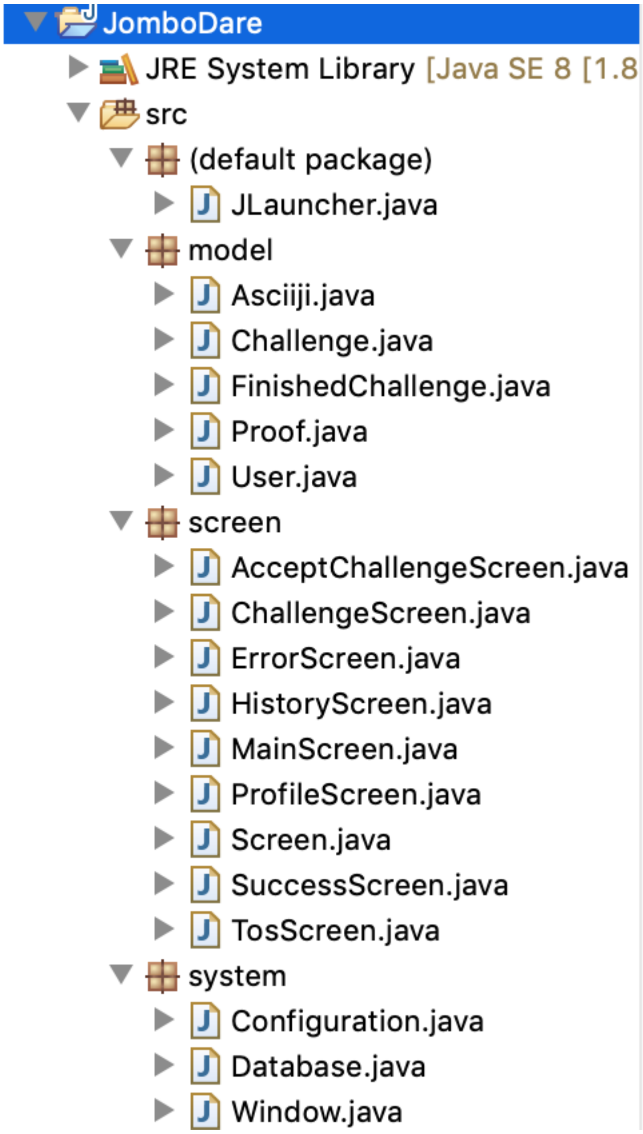


Assembly of the Wingoneer Aluminum Enclosure Case

We then assemble the case with the Pi (know that it is fully configured and working).

5. Software Development & Deployment

I use Java as my choice of programming language because it works on multiple platforms (including Raspbian which runs on a variant Linux Kernel) as well as the JavaFX package to generate GUI windows. I used Eclipse (Oxygen.3A) as my IDE and began working on the program. I designed the program with modularity in mind. Any one component can be added/removed/altered very easily.



```
▼ JomboDare
  ▶ JRE System Library [Java SE 8 [1.8
  ▼ src
    ▼ (default package)
      ▶ JLauncher.java
    ▼ model
      ▶ Asciiji.java
      ▶ Challenge.java
      ▶ FinishedChallenge.java
      ▶ Proof.java
      ▶ User.java
    ▼ screen
      ▶ AcceptChallengeScreen.java
      ▶ ChallengeScreen.java
      ▶ ErrorScreen.java
      ▶ HistoryScreen.java
      ▶ MainScreen.java
      ▶ ProfileScreen.java
      ▶ Screen.java
      ▶ SuccessScreen.java
      ▶ TosScreen.java
    ▼ system
      ▶ Configuration.java
      ▶ Database.java
      ▶ Window.java
```

Packages

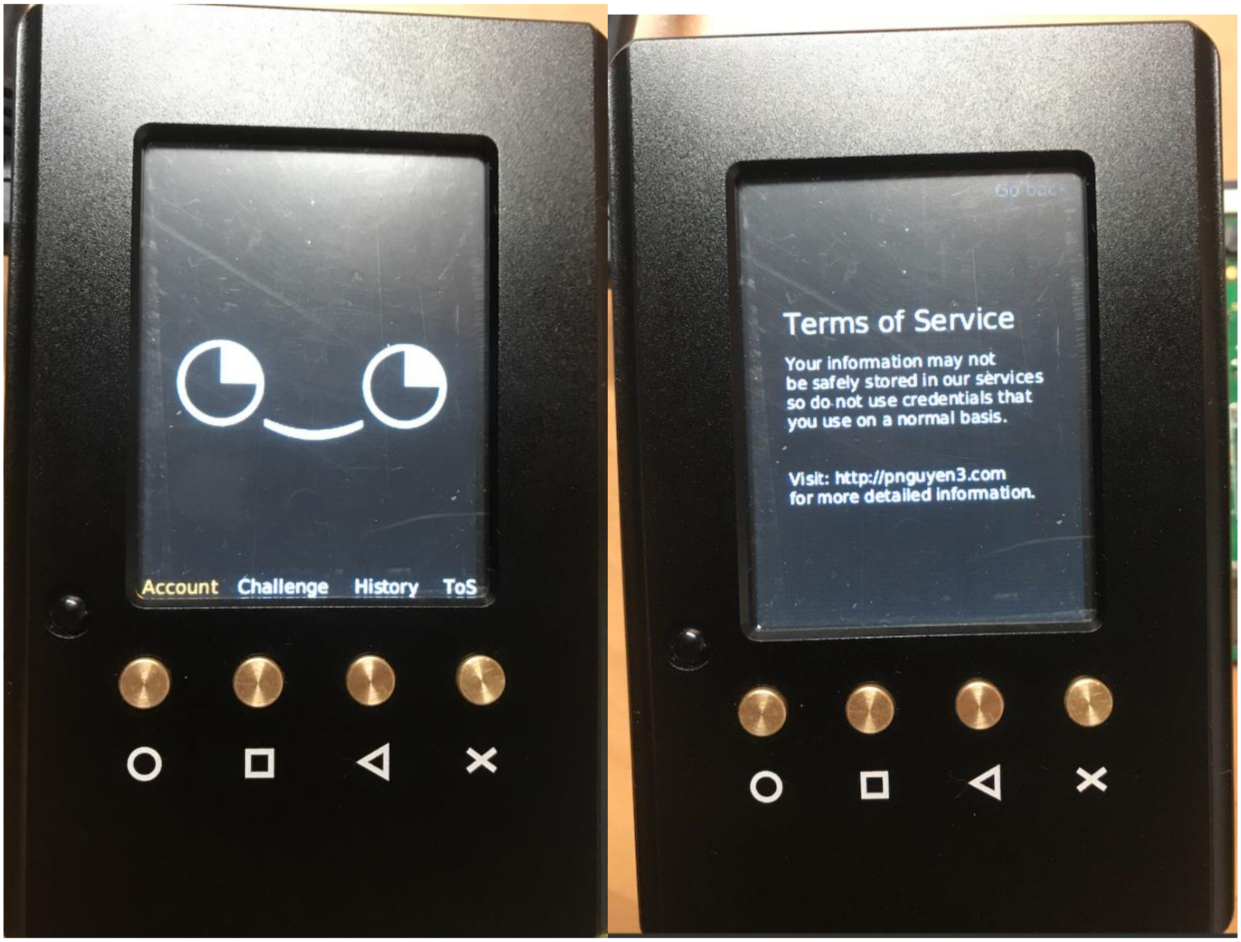
(default package): contains driver file

model: contains all the necessary models to interact with the database. Asciiji is the faces text-database and emotes on the main screen.

screen: contains all the different windows used on the software platform

system: contains all the internal files including GPIO button mappings and connectivity credentials/settings.

The program also runs a shell command ‘sudo mobprobe uinput’ and ‘sudo python kbrd.py’ upon start that will link the GPIO buttons to the mappings on the python script (accordingly to my Java application). It’s run with fullscreen and mouse cursor hidden enabled. The resolution is 240x320 (LCD rotated 180).

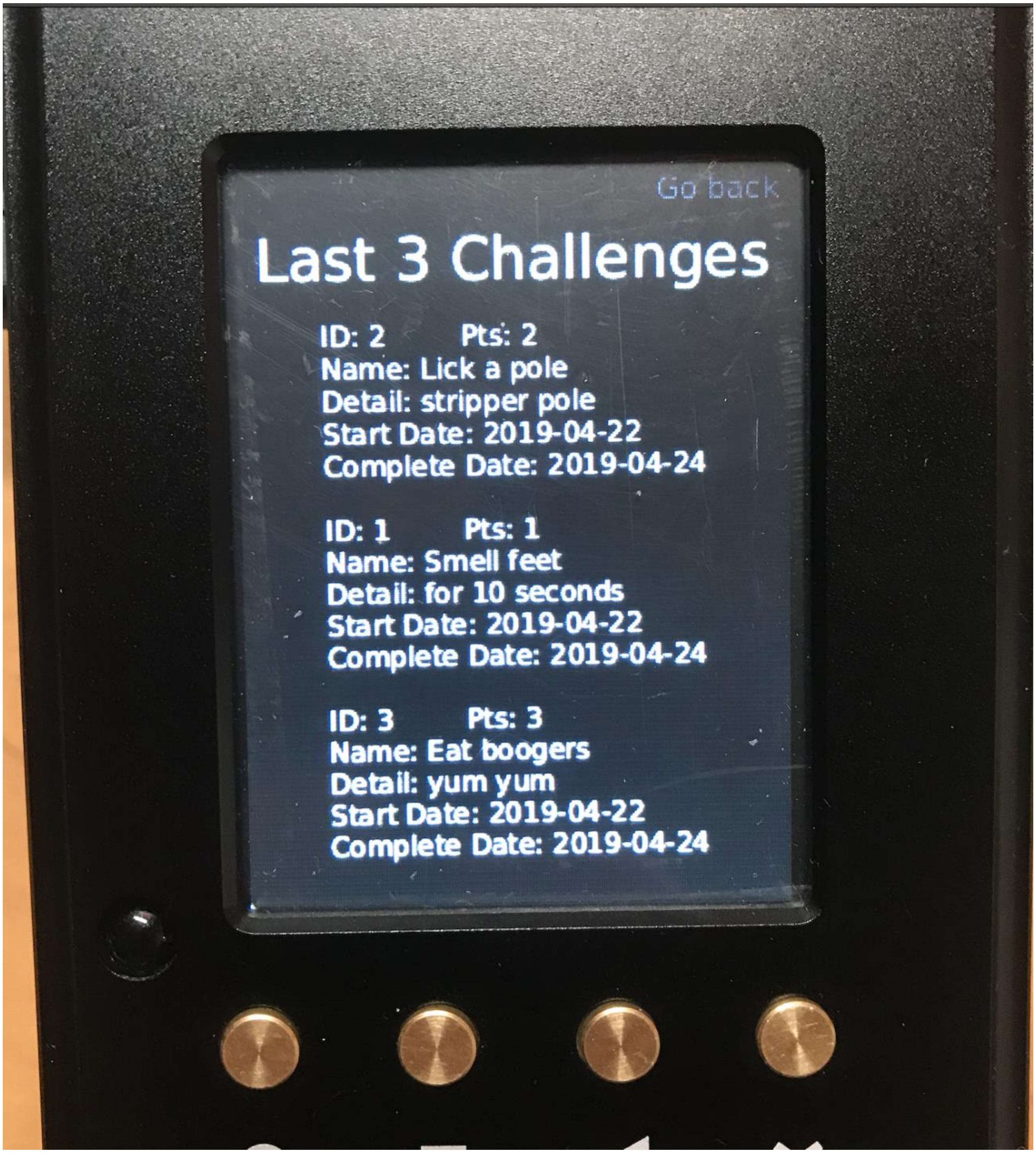


Terms of Service

Your information may not be safely stored in our services so do not use credentials that you use on a normal basis.

Visit: http://pnguyen3.com for more detailed information.

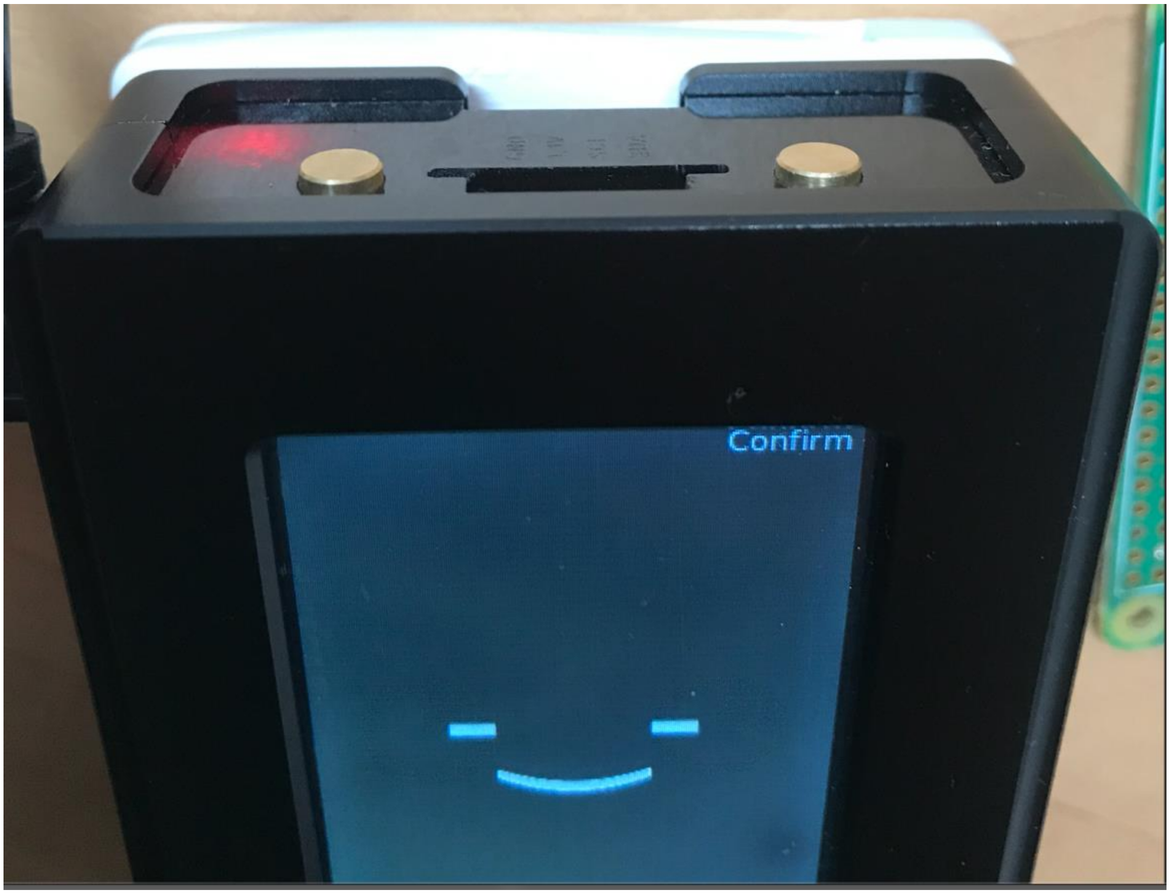
Picture of the application deployed on the Raspberry Pi 3B.



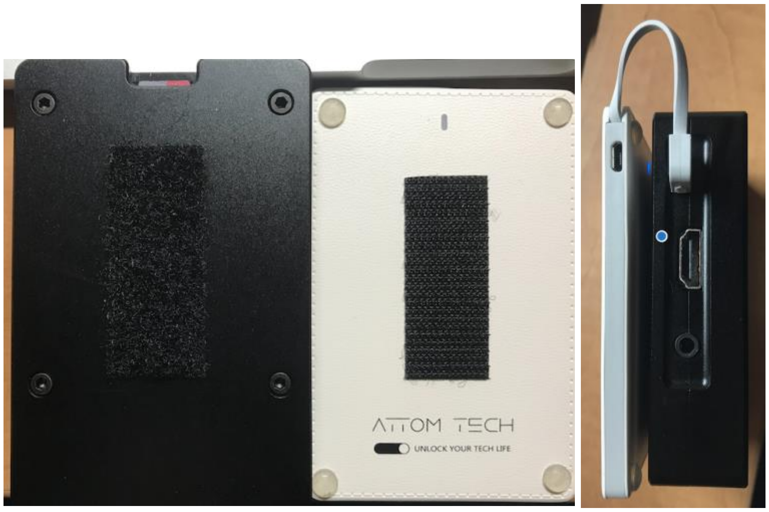
Last 3 Challenges

```
ID: 2        Pts: 2
Name: Lick a pole
Detail: stripper pole
Start Date: 2019-04-22
Complete Date: 2019-04-24

ID: 1        Pts: 1
Name: Smell feet
Detail: for 10 seconds
Start Date: 2019-04-22
Complete Date: 2019-04-24

ID: 3        Pts: 3
Name: Eat boogers
Detail: yum yum
Start Date: 2019-04-22
Complete Date: 2019-04-24
```

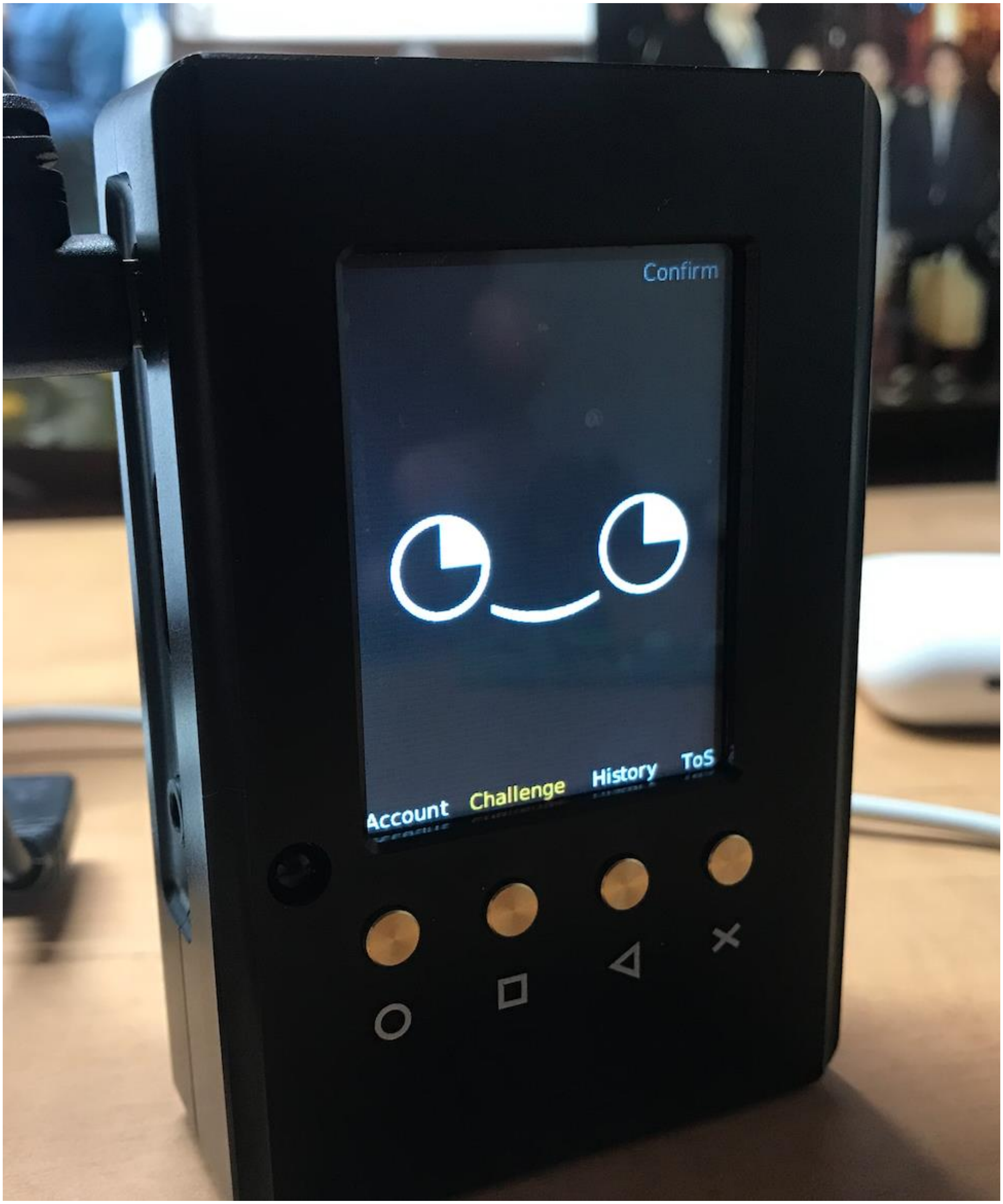
Last 3 challenges completed shown on the 2.2" TFT.



The top button on the right is used to confirm choices.



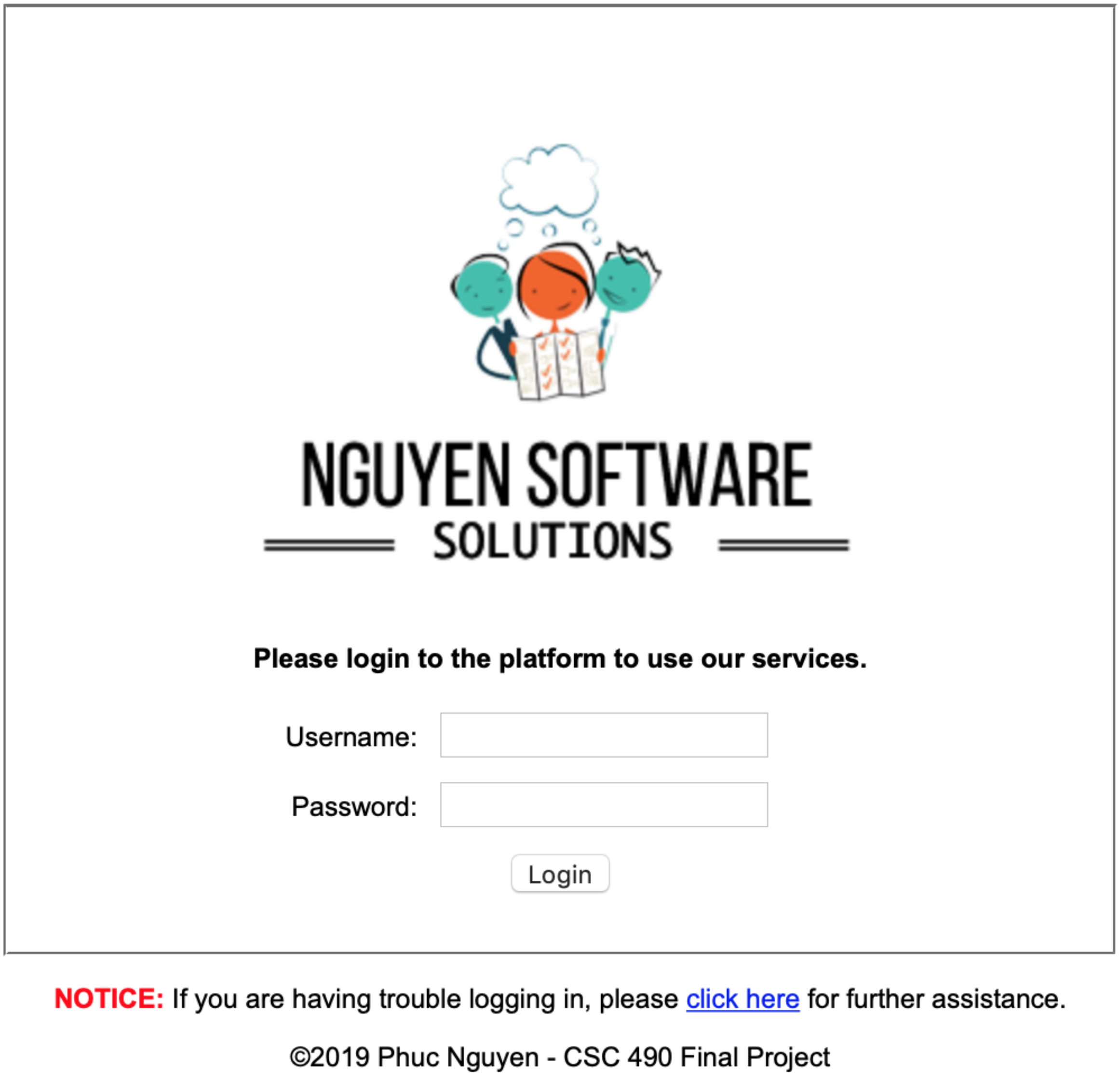
Adhesion of Power Bank to the Raspberry Pi



Completed software deployment on embedded device.

6. Web Deployment

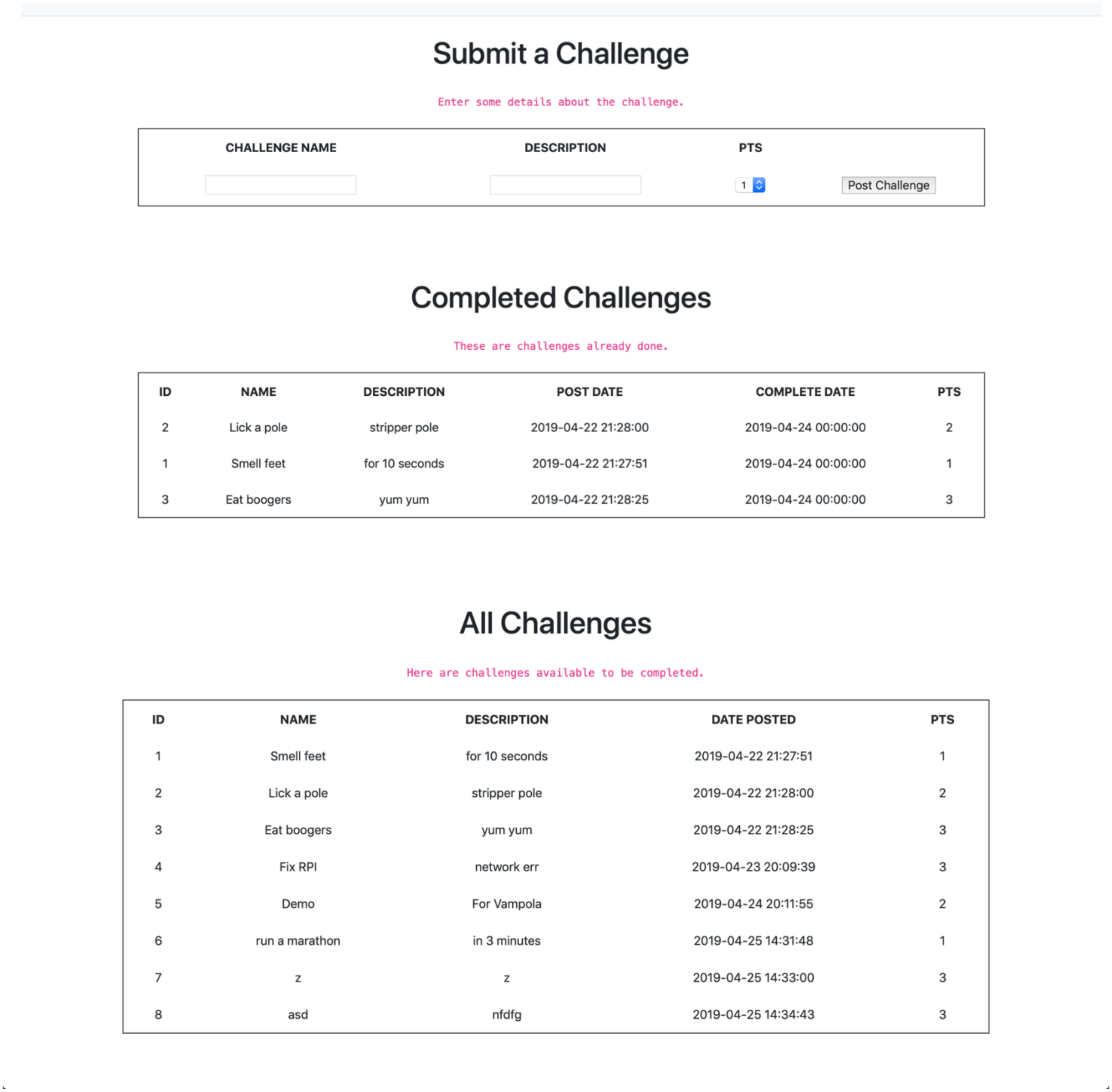
The last part was to deploy a web interface to view challenges completed and posted as well as manage these challenges. Project URL: http://pi.cs.oswego.edu/~pnguyen3/CSC490/



Please login to the platform to use our services.

NOTICE: If you are having trouble logging in, please click here for further assistance.

©2019 Phuc Nguyen - CSC 490 Final Project



Submit a Challenge

Enter some details about the challenge.

CHALLENGE NAME	DESCRIPTION	PTS	
		1	Post Challenge

Completed Challenges

These are challenges already done.

ID	NAME	DESCRIPTION	POST DATE	COMPLETE DATE	PTS
2	Lick a pole	stripper pole	2019-04-22 21:28:00	2019-04-24 00:00:00	2
1	Smell feet	for 10 seconds	2019-04-22 21:27:51	2019-04-24 00:00:00	1
3	Eat boogers	yum yum	2019-04-22 21:28:25	2019-04-24 00:00:00	3

All Challenges

Here are challenges available to be completed.

ID	NAME	DESCRIPTION	DATE POSTED	PTS
1	Smell feet	for 10 seconds	2019-04-22 21:27:51	1
2	Lick a pole	stripper pole	2019-04-22 21:28:00	2
3	Eat boogers	yum yum	2019-04-22 21:28:25	3
4	Fix RPI	network err	2019-04-23 20:09:39	3
5	Demo	For Vampola	2019-04-24 20:11:55	2
6	run a marathon	in 3 minutes	2019-04-25 14:31:48	1
7	z	z	2019-04-25 14:33:00	3
8	asd	nfdfg	2019-04-25 14:34:43	3