Project FAST Technical Design Document

CPT Grayson Osborne

August 27, 2023

1 Introduction

1.1 Background

One of the most formative experiences of a young artilleryman's career is completing manual gunnery instruction during Field Artillery Basic Officer Leader's Course (BOLC) or Advanced Leader's Course (ALC, attended by recently promoted Staff Sergeants). This portion of the course culminates in the "safety test"; a measure of whether or not you have enough understanding of artillery operations to safely operate a Fire Direction Center (FDC) or howitzer gun line.

These exams (and practice in preparation for the exam) are currently created and graded by hand. Although the intent of instruction is to *solve* manual gunnery calculations by hand that does not mean the creation and grading of worksheets needs to be done by hand. Additionally, completing practice and exams through a digital system like a web application creates the potential to capture a great deal of data that may be valuable in honing the instruction of manual gunnery and performance of the students.

The goal of this web application is to improve the process and decrease the workload of instructing manual gunnery at the United States Army Field Artillery School (USAFAS) and to provide Master Gunners in the force the ability to monitor training and performance on semiannual required recertifications.

1.2 Scope of this Document

This document is intended to provide an understanding of the projects, architecture, features, design, and other technical considerations for the project in its production. This project is being developed by one individual which is a significant factor involved in many of the decisions discussed below.

2 System Architecture

2.1 Components, Modules, and Interactions

The application is divided into the following main components:

2.1.1 Front-end Interface

The front-end component will be powered by React and is responsible for rendering the user interface and facilitating user interactions with the database. It includes all screens, forms, and visual elements which the user will interact with directly.

2.1.2 Back-end Services

The back-end services are powered by the Python framework, Django, and handle all of the logic, data processing, and interactions with the database.

2.1.3 Database

The database component of the project will store all the required data for operations.

Much like the back end services, the database tables will likely expand or be altered during the course of development as different needs arise. The database will be normalized to minimize duplicate info and decrease the workload of maintaining the database. This section will include the database schema for this project as it is developed and will also be available as a picture within the git repository.

3 Technologies and Tools

The following tools will be used for each service:

- Front-End Services. The front end services will be provided by React. React allows for the development of fast, responsive, component based applications.
- Back-End Services. The back end services will be provided by Django. Django is a low-code, python back end framework that allows for reliable web applications and comes pre-loaded with many useful features such as user authentication.
- Database. The database service will be provided by Postgres. Postgres is an open-source relational database that includes many helpful features related to performance, security, programming extensions, and configurations.

4 Data Flow and Interaction

4.1 Data Flow Diagram

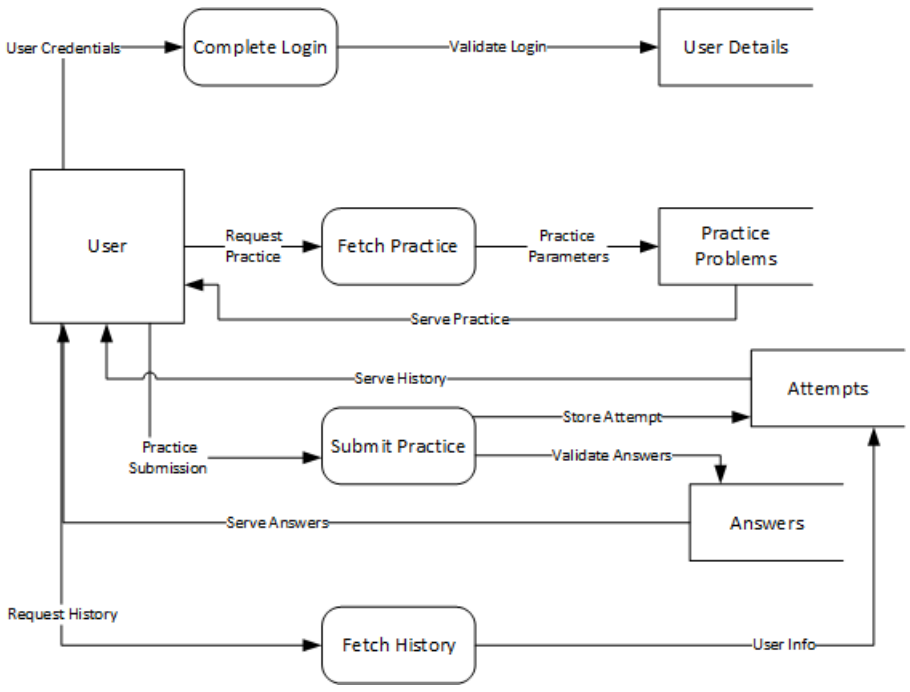

Figure 1: Level 0 Data Flow Diagram

4.2 User Flow Diagram

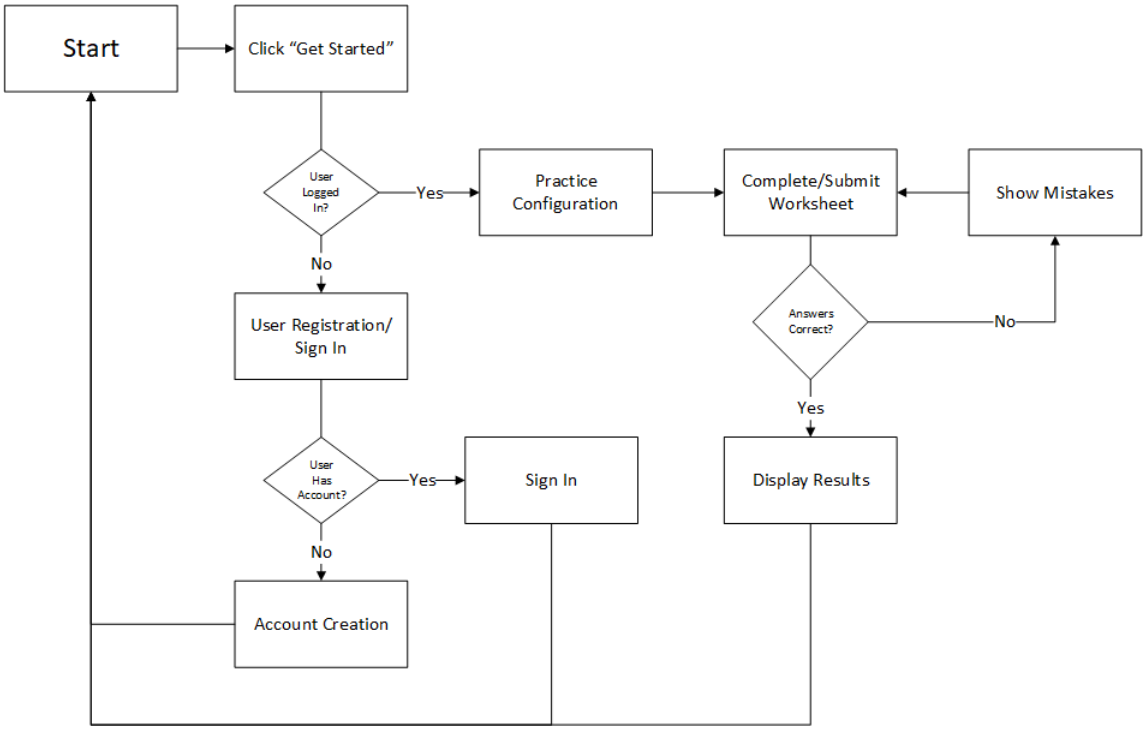

Figure 2: User Flow Diagram

5 Database Schema

I do not currently have enough fidelity on the necessary tables of the database to make it worth creating a database schema diagram. As the project progresses, I will add a database diagram once it becomes clearer.

6 Deployment and Infrastructure

There is currently not a realistic reliable deployment for this web application. The presence of data from a Tabular Firing Table within the database will make it impossible to host independently on any standard commercial web hosting service. When I reach the user testing stage I will work with the Army to see if there are any options for hosting within a network with the requisite compliance to host real users.

7 Minimum Viable Product

The definition of the minimum viable product is extremely important to the success of this project. Determining a valid minimum viable product creates direction and priority for the development of this project and prevents the priorities from being dragged away from the main drive of the project. Although there is a constant draw to implement additional features that aren't strictly necessary but improve performance or user experience, the thrust of the development must always be towards accomplishing the requirements set out in the minimum viable product. Once the minimum viable product is accomplished, we can prioritize and develop features that were bypassed during initial development.

The minimum viable product will consist of the following key features:

- User Account Creation, Authentication, and login.
  Users will be able to register for an account (using .mil domain e-mail). Users will set a username and password. Users will be asked for basic demographic information for future analytics tracking on the platform. Users will be able to login to the website using the account they have created.

- User Settings and Account Information.
  Users will be able to view their settings and account information as well as make updates to some settings.

- Manual Gunnery Problem Generator.
  The manual gunnery problem generator will be able to generate gunnery problems resulting in presenting the user with the initial information sheet including grids, ranges, left and right limits, and round and fuze information.

- Manual Gunnery Problem Solver.
  The manual gunnery problem solver will be able to solve the generated gunnery problems. The solver will generate the solution at problem generation time and will be used to grade the user's input to the practice problems at form submission.

- Manual Gunnery History.
  The manual gunnery history will show each user the problems they have completed. At a minimum, the history will show the time, number of attempts, and test id.

The user stories and feature documentation can be found in the UI Design document. The descriptions above serve as specific benchmarks to be met for the project to be considered to have met the benchmark of minimum viable product.

8 Conclusion

This document serves as the basis of the technical choices for this project. Although it is not exhaustive, as I begin contributing more code to this project I find it important to lay out some of the important details in order to keep focused on some of the more important aspects of development. I will be adding a UI Design document in the coming weeks which will cover the user stories, add more fidelity to feature priority and the minimum viable product. As I continue to work and add more detail into this project this will be a living document I will add more detail to when appropriate.

A Dependencies & Third-Party Libraries

Below are a list of the dependencies and third-part libraries being used within the project.

Django version 4.2
React 18
Postgres version 15.4

As packages are added within these services, the third party libraries and dependencies will be added here.