Problem Statement and Goals
Software Engineering

Team 1, BANDwidth
Declan Young
Ben Dubois
Nathan Uy
Aidan Mariglia

September 24, 2024

Table 1: Revision History

Date	Developer(s)	Change
September 24, 2024	Declan Benjamin Nathan Aidan	Completed problem statement and goals
April 02, 2025	Declan Benjamin Nathan Aidan	Revised problem statement and goals

1 Problem Statement

1.1 Problem

Battery state of charge (SOC) estimation is challenging, requiring specialized algorithms. Standardized testing is necessary to determine the accuracy and performance of proposed approaches. ~~Building off of an existing solution for running SOC algorithms against a test suite, a cloud-based platform will be developed to streamline accepting and running user-submitted algorithms~~. The current solution that runs SOC algorithms against a test suite has a convoluted submission process, which fails regularly, and executes the tests in serial.

1.2 Inputs and Outputs

Inputs:
User developed algorithms, either as matlab ~~or python~~ code.

Outputs:
Algorithm performance or resulting error, history of user submissions, rankings of algorithm performance.

1.3 Stakeholders

- Dr. spencer smith
- Dr. Phil Kollmeyer
- Developers of the project
- Battery Engineering Research students
- Students taking battery engineering class

1.4 Environment

The system requires both software and hardware components to function effectively. The following sections describe the specific software and hardware requirements.

1.4.1 Software

The software environment of the system will be Docker containers built for each component.

1.4.2 Hardware

The docker containers will be run on ~~AWS, either ECS or EKS~~ a local computer. The hardware details of this service are abstracted away from the user.

2 Goals

- The final product can test multiple algorithm submissions in parallel - since each algorithm takes approximately an hour to complete, running them in parallel will help reduce the overall time required to run all algorithms.
- The final product has a user-friendly interface that simplifies the submission process and testing of algorithms.

- The final product generates reports and compares algorithm performance ~~- the reports will summarize the performance of the submitted algorithm and compare it with other submitted algorithms~~, helping users understand and visualize their results.
- The final product can handle any error encountered throughout the algorithm's runtime - this will minimize the software's crashes and downtime and overall, it will increase the reliability of the software. It will make debugging much easier and as a result, improve the user experience.
- The final product is secure and prevents malware attacks - secure software is necessary to protect users' ~~sensitive data~~ submissions and personal information as well as keep the software's integrity and availability.

3 Stretch Goals

Python Implementation of SOC Estimation Algorithm Testing Tool: Currently this tool is Matlab-based and this programming language is more niche and is mainly used in academic roles. For future development, it would be beneficial to have the testing tool and backend written in a programming language like Python that is widely known and used. Also, Matlab is not open-source and requires a license to be used, so translating to the open-source programming language of Python will allow this testing tool to be accessible to everyone.

4 Challenge Level and Extras

The expected challenge level for this project is general. This challenge level is justified firstly by the fact that the technologies and domain knowledge required based on the expected implementation are relatively simple, especially when considering the experience that our group has with web applications and cloud development. Additionally, the problem being solved is not particularly novel, since although the problem itself isn't something that is very commonly solved, the requirements/expected solution of a ~~cloud-based~~ web application is something that is a very common solution to problems in the software industry.

As an extra for the project we will include thorough user guides/walkthroughs as a part of the final webapp, similar to the way github presents their tutorials. Also, a hosting cost analysis will be included that compares different hosting providers.

Appendix — Reflection

Nathan Uy

1. Writing the problem statement allowed me to understand the scope of the project and its impact on stakeholders.

2. Coming up with the extras for the project was challenging as it needed to be both useful and fun. So it required going through a lot of options and evaluating each of them.

Declan Young

1. When writing this deliverable I thought that our group did a great job on collaborating to agree on what the problem that we are solving is, so that the details could be described in a clear and concise way. Since the rest of this deliverable depended on and was based on the problem, it was crucial that we all had the same understanding of the problem.

2. The main pain point we encountered for this deliverable was determining what stretch goals would be most suitable for our project. We resolved this by brainstorming as a group to create a list of possible stretch goals, from which we picked the goal that seemed most reasonable while staying interesting and closely related to the problem being solved.

Aidan Mariglia

1. Defining the problem statement happened quite easily while writing this deliverable. The existing project pitch, along with the information we gained while meeting with Dr. Kollmeyer provided us the necessary background to define the problem well.

2. Determining stretch goals/extras was the worst sticking point of this deliverable. The existing problem was well contained, which added to this difficulty.

Benjamin Dubois

1. During this deliverable, I found that the team was on the same page for all of the required fields and this helped us to complete the deliverable as efficiently as possible. When brainstorming goals and the problem statement, we were able to make decisions very quickly as we all had the same idea of the problem and main goals.

2. The main pain point that we encountered during this deliverable was determining the stretch goals and extras. This was difficult as some features are hard to determine how long they will take at this moment, so it is difficult to know which features we can easily include in the timeline and

which ones will need to be stretch goals. To resolve this, we created a list of the major development features we wanted to complete, and any features that seemed less significant were added to stretch goals or extras.

Team

3. We adjusted our project's scope to make it suitable for a Capstone project by firstly determining the scope of that project required to complete the project at the most basic level. We then added to this scope to make the project more complex and the scope more broad, until a point was reached where we felt that any more additions would be overwhelming and difficult to complete with the given time and resources. We increased the scope by adding extras to our project, as well as adding stretch goals that we hope to complete.