

Reflection Report on MCT: A Command Scheduling Application for Mission Operation and Control (MOC) of the McMaster PRESET CubeSat

1 Changes in Response to Feedback

Summarize the changes made over the course of the project in response to feedback from TAs, the instructor, teammates, other teams, the project supervisor (if present), and from user testers.

For those teams with an external supervisor, please highlight how the feedback from the supervisor shaped your project. In particular, you should highlight the supervisor's response to your Rev 0 demonstration to them.

1.1 SRS and Hazard Analysis

The SRS has been modified to include a more accurate diagram about system boundaries as this caused some ambiguity on the inputs and outputs of the system. Formal math describing the three main flows between a user, satellite and command schedule was written. Reduced ambiguity and was more clear in parts of the SRS by stating if certain things will be considered, implemented etc. Included more labelling, more descriptions for personas, consistent terminology used for all requirements.

1.2 Design and Design Documentation

The module interface specification documentation has been modified to include syntax for previously ambiguous functions. Details regarding functions inputs, outputs and their formats has also been added. Furthermore, formal math has been included to functions where required. Additionally, the reflection section is divided into questions and answers those questions in a detailed manner that delves into multiple aspects of the project including design and validation. Changes made to the module guide include unique identifiers for tables, clearer traceability for modules and improvised diagram layouts.

1.3 VnV Plan and Report

The VnV Plan has been modified to include specific details pertaining to the validation and nondynamic testing of the application. It also now further explains how information gathering methods such as checklists and surveys had been used to elicit user needs.

The VnV Report has been updated to encompass the verification of the non-functional requirements, performance and usability. Data in graphical form has been added to support the system tests pertaining to performance and usability. Additionally, further analysis and conclusions about the project has been made based on the validation data, where it mentions the qualities and areas that have been improved.

2 Design Iteration (LO11)

Reflecting on the project from its conception to the final product, the iterative process of designing, requirements gathering, implementing, testing, and evaluating the application has been pivotal in shaping the final version of the product. This process guided us through a continuous loop of refinement, allowing us to adapt and evolve our project in response to feedback and testing outcomes.

The biweekly meetings that we held with our stakeholders greatly contributed in the iterative process of the design. In the early stages of the project, we used these meetings to discuss the requirements and software specifications during the design phase. These meetings ensured that the progression of the project remained aligned with the stakeholders' needs. As we began implementing the application, these biweekly meetings were used to gather feedback on our progress, which helped make improvements for the subsequent iterations. In addition, these meetings also helped clarify functional requirements as we began implementing them. For example, one of the major tasks involved creating a flow for scheduling a list of commands for a specific overpass. There were concerns regarding the automation of executing these schedules as an overpass for a satellite system occurs. These meetings allowed us to present and discuss approaches to such problems that would better align with their use-cases. This constant engagement helped refine the requirements specification and ensured that their insights and preferences were integral to the development process.

After the initial implementation of the application, we conducted two usability tests which helped us iterate the design of the user interface. The insights gained from each usability testing round highlighted areas where our design could be more intuitive and user-friendly, leading to targeted iterations focused on enhancing the UI/UX. Following each usability testing demonstration, the feedback gathered was used to reiterate the design of the frontend interface such that it was more intuitive, accessible, and efficient for users.

Furthermore, we continuously switched between implementing and testing our product to make sure that the development or refinement of features were tested within the context of the whole application. This helped us evaluate the impact of these changes on the overall functionality and usability of the application. This iterative process helped steer us towards a final design that matches with the project's goals.

3 Design Decisions (LO12)

Reflecting on the design decisions made throughout the course of this project, the assumptions, limitations, and constraints established in the early design stages helped influence our choices.

3.1 Assumptions

The assumptions for this project influenced our selection of technologies and tools used to build the application. Firstly, one of the assumptions was that the application would not be hosted beyond a single server. Knowing this, we were able to direct our attention designing the application to be lightweight and efficient, optimizing performance to fit within the limitations of a single-server architecture. Secondly, another assumption was that the application would not store any science data collected by the satellite. This eliminated the need for large-scale data storage solutions and complex data management systems. Instead, we were able to make the design decision to focus more time on handling command and control operations, which helped streamline the application's functionality to serve the primary purpose of scheduling satellite commands effectively. Lastly, the assumption that the application is not responsible for designing commands internally also helped focus development efforts. This allowed us to focus on building a command scheduler that is robust with various types of commands.

3.2 Limitations and Constraints

The limitation and constraints specified in the requirements specification dictated the selection of certain technologies. One of the constraints was that the application should use the OpenID Connect Protocol to authenticate users. This required us to integrate authentication framework which is compliant with this protocol. Another constraint was that two line element (TLE) data needed to be used to predict a satellite's overpass and illumination cycle. This constraint led us to make the decision to use a verified, external package to process and utilize TLE data, ensuring that the application could perform these critical calculations with up-to-date information.

4 Economic Considerations (LO23)

Given the project's aim to reduce friction and increase efficiency for any satellite group to send commands from their ground station to their satellite, there is a niche market for the product. However, the nature of the aerospace industry is built on the back of trust, and open-source software - as seen from the plethora of repositories available for the public from organizations such as NASA, and the Canadian Space Agency. As such, the product would also be open-sourced, rendering it accessible for organizations of any size to use and to benefit from.

The estimated cost to produce a version that would be desirable to use would increase significantly, from server hosting costs to database costs, as well as further development for a product which is faster, and more responsive.

This will, in turn, attract users who are looking for low cost solutions to improving the throughput of their satellite scheduling capacity. As the aerospace industry is quite small (from group members professional experiences), word of mouth would be the best form of marketing this product, including the NASA public mailing group, contacting small satellite organizations worldwide, and potentially demonstrating the product at various space events.

This word of mouth marketing is starkly different than commercial marketing - since our market is niche and well connected, it would be a loss to market to the general public, for which the majority would have no use or interest in the product. Staying closer to our users would be more beneficial to increase the number of users.

There are approximately 3000 satellites currently in orbit, with an estimated potential 3000 users for our product.

5 Reflection on Project Management (LO24)

This question focuses on processes and tools used for project management.

5.1 How Does Your Project Management Compare to Your Development Plan

Did you follow your Development plan, with respect to the team meeting plan, team communication plan, team member roles and workflow plan. Did you use the technology you planned on using?

With respect to workflow plan, team member roles and communication plan, we did follow our Development Plan sufficiently well. We stated in our Development plan that we will use our Lower Earth Orbiters server on Discord to communicate which is what we did throughout the course of both terms. Ad-

ditionally, the Development Stack and the technologies mentioned in the Dev Plan were what we relied on in making this application come to life. We ensured that each team member's role and responsibilities were maintained by reminding members of the team to do tasks that fit more of their role. We were very consistent in team meetings and coordinating a time that works for everyone, hence our overall team meeting attendance was quite high. Overall, the project management was done well.

5.2 What Went Well?

What went well for your project management in terms of processes and technology?

In terms of project management, communication between team members was always of high quality. Having regular meetings over Discord to discuss ideas, code reviews, sharing a screen when running into errors were all good techniques we employed to ensure things were moving. Enabling extensions such as ESLint into our development environments made it easy to maintain a high level of code quality and readability.

5.3 What Went Wrong?

What went wrong in terms of processes and technology?

The only limitation we faced when building our application is the constraints of following a fixed development stack. There were solutions that we could have incorporated at key times in our software development process where it would involve different libraries or tools that we was not in our original plan. This was a common problem we faced. Additionally, the satellite and other astronomical calculations were difficult, outsourcing this to a library made it easier.

5.4 What Would you Do Differently Next Time?

What will you do differently for your next project?

For our next project, we would all try to take on different roles so that each member gets to try something new. This allows each member to take on a new challenge and learn different aspects of software development. This is important because it's easy to be comfortable with what you are good at and like, but switching roles and responsibilities can offer something new to each team member. Additionally, a different development stack can be interesting and fun because it always poses a new challenge and learning opportunity. While our stack was robust, we believe experimenting with newer frameworks or languages could offer improved performance and developer productivity. Finally, regarding user experience, gathering more user feedback through usability testing and

incorporating UX/UI design principles more thoroughly could have significantly improved the application's usability and overall user satisfaction in hindsight.