

Software Requirements Specification for  
McMaster Engineering Society Custom  
Financial Expense Reporting Platform:  
subtitle describing software

Team #12, Reimbursement Rangers

Adam Podolak

Evan Sturmey

Christian Petricca

Austin Bennett

Jacob Kish

October 11, 2024

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>vi</b>
1.1	User Business . . . . .	vi
1.2	Goals of the Project . . . . .	vi
<b>2</b>	<b>Stakeholders</b>	<b>vi</b>
2.1	Client . . . . .	vi
2.2	Customer . . . . .	vi
2.3	Other Stakeholders . . . . .	vi
2.4	Hands-On Users of the Project . . . . .	vii
2.5	Personas . . . . .	vii
2.6	Priorities Assigned to Users . . . . .	vii
2.7	User Participation . . . . .	vii
2.8	Maintenance Users and Service Technicians . . . . .	vii
<b>3</b>	<b>Mandated Constraints</b>	<b>viii</b>
3.1	Solution Constraints . . . . .	viii
3.2	Implementation Environment of the Current System . . . . .	viii
3.3	Partner or Collaborative Applications . . . . .	viii
3.4	Off-the-Shelf Software . . . . .	viii
3.5	Anticipated Workplace Environment . . . . .	viii
3.6	Schedule Constraints . . . . .	ix
3.7	Budget Constraints . . . . .	ix
3.8	Enterprise Constraints . . . . .	ix
<b>4</b>	<b>Naming Conventions and Terminology</b>	<b>ix</b>
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project . . . . .	ix
<b>5</b>	<b>Relevant Facts And Assumptions</b>	<b>ix</b>
5.1	Relevant Facts . . . . .	ix
5.2	Business Rules . . . . .	x
5.3	Assumptions . . . . .	x
<b>6</b>	<b>The Scope of the Work</b>	<b>x</b>
6.1	The Current Situation . . . . .	x
6.2	The Context of the Work . . . . .	x
6.3	Work Partitioning . . . . .	xi

6.4	Specifying a Business Use Case (BUC)	xi
<b>7</b>	<b>Business Data Model and Data Dictionary</b>	<b>xi</b>
7.1	Business Data Model	xi
7.2	Data Dictionary	xi
<b>8</b>	<b>The Scope of the Product</b>	<b>xi</b>
8.1	Product Boundary	xi
8.2	Product Use Case Table	xi
8.3	Individual Product Use Cases (PUC's)	xi
<b>9</b>	<b>Functional Requirements</b>	<b>xi</b>
9.1	Functional Requirements	xi
<b>10</b>	<b>Look and Feel Requirements</b>	<b>xii</b>
10.1	Appearance Requirements	xii
10.2	Style Requirements	xii
<b>11</b>	<b>Usability and Humanity Requirements</b>	<b>xii</b>
11.1	Ease of Use Requirements	xii
11.2	Personalization and Internationalization Requirements	xii
11.3	Learning Requirements	xii
11.4	Understandability and Politeness Requirements	xii
11.5	Accessibility Requirements	xii
<b>12</b>	<b>Performance Requirements</b>	<b>xii</b>
12.1	Speed and Latency Requirements	xii
12.2	Safety-Critical Requirements	xiii
12.3	Precision or Accuracy Requirements	xiii
12.4	Robustness or Fault-Tolerance Requirements	xiii
12.5	Capacity Requirements	xiii
12.6	Scalability or Extensibility Requirements	xiii
12.7	Longevity Requirements	xiv
<b>13</b>	<b>Operational and Environmental Requirements</b>	<b>xiv</b>
13.1	Expected Physical Environment	xiv
13.2	Wider Environment Requirements	xiv
13.3	Requirements for Interfacing with Adjacent Systems	xiv
13.4	Productization Requirements	xiv

13.5 Release Requirements . . . . .	xv
<b>14 Maintainability and Support Requirements</b>	<b>xv</b>
14.1 Maintenance Requirements . . . . .	xv
14.2 Supportability Requirements . . . . .	xv
14.3 Adaptability Requirements . . . . .	xvi
<b>15 Security Requirements</b>	<b>xvi</b>
15.1 Access Requirements . . . . .	xvi
15.2 Integrity Requirements . . . . .	xvi
15.3 Privacy Requirements . . . . .	xvi
15.4 Audit Requirements . . . . .	xvi
15.5 Immunity Requirements . . . . .	xvi
<b>16 Cultural Requirements</b>	<b>xvii</b>
16.1 Cultural Requirements . . . . .	xvii
<b>17 Compliance Requirements</b>	<b>xvii</b>
17.1 Legal Requirements . . . . .	xvii
17.2 Standards Compliance Requirements . . . . .	xvii
<b>18 Open Issues</b>	<b>xvii</b>
<b>19 Off-the-Shelf Solutions</b>	<b>xviii</b>
19.1 Ready-Made Products . . . . .	xviii
19.2 Reusable Components . . . . .	xviii
19.3 Products That Can Be Copied . . . . .	xix
<b>20 New Problems</b>	<b>xix</b>
20.1 Effects on the Current Environment . . . . .	xix
20.2 Effects on the Installed Systems . . . . .	xix
20.3 Potential User Problems . . . . .	xx
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xx
20.5 Follow-Up Problems . . . . .	xx
<b>21 Tasks</b>	<b>xxi</b>
21.1 Project Planning . . . . .	xxi
21.2 Planning of the Development Phases . . . . .	xxi

<b>22 Migration to the New Product</b>	<b>xxi</b>
22.1 Requirements for Migration to the New Product . . . . .	xxi
22.2 Data That Has to be Modified or Translated for the New System	xxii
<b>23 Costs</b>	<b>xxii</b>
<b>24 User Documentation and Training</b>	<b>xxii</b>
24.1 User Documentation Requirements . . . . .	xxii
24.2 Training Requirements . . . . .	xxiii
<b>25 Waiting Room</b>	<b>xxiii</b>
<b>26 Ideas for Solution</b>	<b>xxiii</b>

## Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

# **1 Purpose of the Project**

## **1.1 User Business**

The project aims to streamline the process of handling reimbursement requests for the McMaster Engineering Society, facilitating proper budget tracking, improved efficiency and user experience, as well as reducing human error.

## **1.2 Goals of the Project**

Goals include:

- User-friendly reimbursement submission interface.
- Tracking of reimbursement requests.
- Tiered access to the reimbursement platform for different users.
- Invoice generation.

# **2 Stakeholders**

## **2.1 Client**

This software is designed for the McMaster Engineering Society and its staff, and by extension for McMaster University.

## **2.2 Customer**

MES clubs/team members that are required to submit reimbursement requests for their projects and club work.

## **2.3 Other Stakeholders**

University administration, future developers of the finance platform, and outside auditors.

## 2.4 Hands-On Users of the Project

Day to day, the software will be used by students submitting reimbursement requests for themselves or on their teams' behalf, as well as MES finance staff who are to review requests and handle them accordingly.

## 2.5 Personas

- **Club Treasurers:** Needs to be able to submit reimbursement requests easily and frequently, as well as track the status of requests.
- **Finance Staff:** Needs to be able to effectively manage all incoming reimbursement requests from different clubs/users.
- **Club Member/General Students:** Infrequent reimbursement request submission; user-friendliness is required.
- **Auditor:** Needs ready access to necessary financial documentation submitted to and generated by the finance platform.

## 2.6 Priorities Assigned to Users

Highest to Lowest:

1. MES Finance Staff
2. Club Treasurers
3. Club Members/General Students
4. Auditors

## 2.7 User Participation

User testing will be conducted with Club Treasurers/Club Members as well as the MES Finance staff to ensure requirements are met.

## 2.8 Maintenance Users and Service Technicians

MES staff will maintain the system after its deployment.



## **3 Mandated Constraints**

### **3.1 Solution Constraints**

The finance platform will adhere to the data privacy regulations of McMaster University. The finance platform will integrate with existing McMaster IT infrastructure. The finance platform will only be available to McMaster affiliated individuals, including students, team/club members, treasurers, MES finance staff, McMaster Administration, and outside auditors.

### **3.2 Implementation Environment of the Current System**

The current system operates using a combination of Google Forms submissions and Google Sheets for tracking. MongoDB is used for database management, and Digital Ocean is used for cloud management.

### **3.3 Partner or Collaborative Applications**

While the finance platform will involve a technology stack, it will be standalone in its operations. It will integrate with existing MES IT infrastructure but will wholly replace the existing reimbursement platform.

### **3.4 Off-the-Shelf Software**

Some existing off-the-shelf solutions include Intuit QuickBooks and Xero. QuickBooks could be used for ledger tracking and invoice generation, while Xero can also streamline the invoice process. It is unclear at this stage which off-the-shelf software will be employed, if any.

### **3.5 Anticipated Workplace Environment**

The finance platform will primarily be accessed online via a website. Mobile access will also be possible. Although access will be restricted to McMaster affiliated individuals, the platform will be accessible from anywhere.

### **3.6 Schedule Constraints**

The finance platform must be completed by the end of the Winter 2025 term; however, completing it sooner is ideal to address existing MES finance concerns.

### **3.7 Budget Constraints**

The project at this stage is not projected to require additional funds.

### **3.8 Enterprise Constraints**

As previously mentioned, the finance platform must adhere to McMaster University's data privacy regulations and cybersecurity standards.

## **4 Naming Conventions and Terminology**

### **4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project**

- **MES:** McMaster Engineering Society
- **IT:** Information Technology
- **UAT:** User Acceptance Testing
- **API:** Application Programming Interface

## **5 Relevant Facts And Assumptions**

### **5.1 Relevant Facts**

The current MES reimbursement process relying on Google Forms and Google Sheets has proven cumbersome, and as a result, students are not inclined to make timely reimbursement requests, introducing avoidable problems to the finance team. The MES does not receive its due fees from McMaster University unless they pass their audits, making the process of submitting reimbursement requests integral to the MES staying operational and receiving

its due funding. Outside of efficiency and availability, the existing platform also does not include enough tracking/updates along the journey of a given request, making some information susceptible to being lost.

## **5.2 Business Rules**

- All reimbursement requests must include a receipt or invoice as proof of purchase.
- The MES Finance team has the authority to modify the reimbursement process as needed to handle extraneous circumstances.
- The platform's functionality is subject to change in the future as per the needs of the MES.

## **5.3 Assumptions**

- Unless otherwise stated, the MES finance team will be able to review requests in a fixed, predetermined timeline.
- Reimbursement requests can only be made with all requisite information and documentation.
- Reimbursement requests will only be made from MES Clubs/Teams to the MES.
- Internet access will be required to use the finance platform.

# **6 The Scope of the Work**

## **6.1 The Current Situation**

*Insert your content here.*

## **6.2 The Context of the Work**

*Insert your content here.*

## **6.3 Work Partitioning**

*Insert your content here.*

## **6.4 Specifying a Business Use Case (BUC)**

*Insert your content here.*

# **7 Business Data Model and Data Dictionary**

## **7.1 Business Data Model**

*Insert your content here.*

## **7.2 Data Dictionary**

*Insert your content here.*

# **8 The Scope of the Product**

## **8.1 Product Boundary**

*Insert your content here.*

## **8.2 Product Use Case Table**

*Insert your content here.*

## **8.3 Individual Product Use Cases (PUC's)**

*Insert your content here.*

# **9 Functional Requirements**

## **9.1 Functional Requirements**

*Insert your content here.*

## **10 Look and Feel Requirements**

### **10.1 Appearance Requirements**

*Insert your content here.*

### **10.2 Style Requirements**

*Insert your content here.*

## **11 Usability and Humanity Requirements**

### **11.1 Ease of Use Requirements**

*Insert your content here.*

### **11.2 Personalization and Internationalization Requirements**

*Insert your content here.*

### **11.3 Learning Requirements**

*Insert your content here.*

### **11.4 Understandability and Politeness Requirements**

*Insert your content here.*

### **11.5 Accessibility Requirements**

*Insert your content here.*

## **12 Performance Requirements**

### **12.1 Speed and Latency Requirements**

- Payment requests would be processed by the system within 5 seconds

- Invoices would be generated within 3 seconds
- Reimbursement requests shall be processed within 5 seconds

## **12.2 Safety-Critical Requirements**

- Sensitive information such as credit card numbers shall be masked
- Access controls shall restrict access to confidential information
- Strong encryption would provide secure data transfer between steps

## **12.3 Precision or Accuracy Requirements**

- Monetary values would be rounded off to the nearest cent (i.e., x.yz)

## **12.4 Robustness or Fault-Tolerance Requirements**

- Backup servers shall store critical information in case of failure, including daily backups
- Graceful error handling would preserve sensitive information in case of improper use

## **12.5 Capacity Requirements**

- The system shall be able to handle 1000 reimbursement requests, 1000 invoices, and 500 payment requests daily
- The system would be able to handle up to 10 requests simultaneously
- The system shall be able to store up to 10 years' worth of backed-up data

## **12.6 Scalability or Extensibility Requirements**

- API integration would support future third-party connectivity
- The system would be sufficiently scalable so as to be able to accommodate double its current maximum usage within one year

## 12.7 Longevity Requirements

- The system shall be serviceable, at its current usage rate, for at least the next 10 years without needing considerable updates or overhauls made to it

## 13 Operational and Environmental Requirements

### 13.1 Expected Physical Environment

- The system shall run on Windows 10 and 11 operating systems
- The system shall require similar minimum specs to Intuit Quickbooks to run

### 13.2 Wider Environment Requirements

- Financial institutions integrated with the payment/reimbursement features shall make use of industry-standard systems and processes for Canadian banks

### 13.3 Requirements for Interfacing with Adjacent Systems

- The system shall be adaptable enough to gracefully handle interactions with different financial institutions
- The API shall be adaptable for use with other (Future) third-party connections

### 13.4 Productization Requirements

- **Scalability and Performance:** Ensure server and database can handle increased usage and data with automated scaling capabilities.
- **Security:** Implement data encryption, secure authentication (e.g., OAuth, 2FA)

- **User Experience:** Develop an intuitive interface with role-based access control (RBAC) and adhere to WCAG accessibility guidelines.
- **Integration with Existing Systems:** Ensure compatibility with MES infrastructure (e.g., DigitalOcean) and integrate with external services (e.g., payment processors, audit tools).
- **Maintainability and Support:** Provide comprehensive documentation, automated testing (CI/CD) via GitHub Actions, and ensure error handling with clear logs and user feedback.
- **Training and Onboarding:** Create training materials for users and provide help desk support during the transition to the new platform.
- **Cost and Future-Proofing:** Monitor infrastructure costs, plan for scalable cloud resource usage, and design the platform to be modular and future-proof.

## 13.5 Release Requirements

- Stable releases shall be released twice a year

# 14 Maintainability and Support Requirements

## 14.1 Maintenance Requirements

- Data backups shall be performed daily to protect sensitive information
- The system shall experience scheduled maintenance downtime for 10 percent of each year (specifics to be determined)
- Security updates and patches shall be released as needed (separate from the twice-yearly stable releases)

## 14.2 Supportability Requirements

- Documentation for users and the team shall be kept up to date as new releases are made



### **14.3 Adaptability Requirements**

- The system shall be robust enough to adapt to changes from external and connected sources such as McMaster University or Canadian financial institutions without losing core functionalities

## **15 Security Requirements**

### **15.1 Access Requirements**

- The system shall ensure that users only have access to the data they are authorized to
- The system shall have a secure login portal to prevent unauthorized access

### **15.2 Integrity Requirements**

- The system shall log 100% of transactions to ensure financial history is traceable

### **15.3 Privacy Requirements**

- The system shall encrypt sensitive information in transit and at rest
- The system shall return anonymous data when generating reports that do not require personal info

### **15.4 Audit Requirements**

- The system shall maintain a detailed log of 100% of transactions to ensure financial transparency
- The system shall retain audit logs for a minimum of 10 years

### **15.5 Immunity Requirements**

- The system shall be resilient to data corruption, ensuring that data can be recovered within 4 hours of a failure

## 16 Cultural Requirements

### 16.1 Cultural Requirements

- The system shall be scaleable to implement future multi-language support to accommodate international students or staff

## 17 Compliance Requirements

### 17.1 Legal Requirements

- The system shall follow user data protection laws to ensure proper handling of sensitive information.
- The system shall follow any relevant financial regulations for non-profit organizations.

### 17.2 Standards Compliance Requirements

- The system shall follow web security standards

## 18 Open Issues

The key open issues currently revolve around undecided functionality, particularly regarding the integration of automated reporting features and the appropriate level of customization for the needs of various student groups. Other considerations include:

- How granular the platform should be in tracking compliance for audit purposes.
- Whether the system will support all financial workflows such as reimbursement requests, intramural funding, and payment tracking.
- Ensuring that the user experience is intuitive for both administrators and students.

These decisions will significantly influence the overall architecture of the solution and the development timeline.

## 19 Off-the-Shelf Solutions

Accounting and invoicing software is readily available, however MES refrains from implementing these existing solutions due to a variety of factors. Commonly cited concerns include:

- Lack of control over platform customization to meet the specific needs of MES.
- High subscription fees that are unsustainable for a non-profit organization.
- Data privacy concerns, especially in handling sensitive financial information.
- Limited support for integration with existing MES infrastructure.

### 19.1 Ready-Made Products

Examples of existing financial management tools that were considered but not implemented include:

- **Wave Accounting** (<https://www.waveapps.com/>) - A free accounting platform suited for small businesses but lacks customization for non-profit needs.
- **FreshBooks** (<https://www.freshbooks.com/>) - A paid tool offering invoicing and expense tracking, but deemed too expensive and restrictive for MES.
- **Xero** (<https://www.xero.com>) - Cloud based accounting platform, for invoicing, inventory and bank reconciliation
- **QuickBooks Online** (<https://quickbooks.intuit.com/>) - Widely-used platform for small to mid sized businesses

### 19.2 Reusable Components

There are numerous reusable components that could be adapted to suit the needs of the MES. These include:

- **LedgerSMB** (<https://ledgersmb.org/>) - An open-source accounting tool that could be modified for more specific MES needs.
- **Plaid API** (<https://plaid.com/docs/api/>) - For integrating bank data and managing student group transactions.
- **Stripe API** (<https://docs.stripe.com/api>) - For handling payments and reimbursements.

### 19.3 Products That Can Be Copied

While no single product fully meets MES requirements, certain features from established platforms could inspire the design of the financial management system:

- **QuickBooks** - Budgeting tools and expense tracking can provide a model for managing student group funds.
- **Expensify** - Reimbursement workflows and receipt tracking features.

## 20 New Problems

### 20.1 Effects on the Current Environment

The implementation of a new centralized platform for financial management may affect MES's current environment in the following ways:

- Increased demand on server resources, especially as more financial processes move from local and manual to online and automated.
- Potential need for server upgrades if the platform requires more CPU, RAM, or storage than the existing DigitalOcean infrastructure provides.

### 20.2 Effects on the Installed Systems

MES currently uses DigitalOcean for its infrastructure. Introducing a financial management platform could result in:

- Increased usage limits, potentially necessitating the purchase of higher-tier virtual machines.
- A larger database capacity for tracking all transactions and handling potentially higher loads during peak reporting periods.

### **20.3 Potential User Problems**

User challenges may arise from:

- The steep learning curve for administrators and student groups transitioning from the current system of Google Forms, PDFs, and spreadsheets to a new, fully automated platform.
- Potential resistance from users who are accustomed to the manual processes and may find it difficult to adapt to a new system.

### **20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Potential limitations include:

- Resource constraints on the DigitalOcean infrastructure, which could limit the performance and scalability of the platform.
- Increased operating costs due to the need for more powerful virtual machines or cloud services.

### **20.5 Follow-Up Problems**

After implementation, follow-up issues could arise, such as:

- Reconciling the codebase of the new platform with existing MES systems and ensuring seamless integration.
- Ensuring that the new platform adheres to financial compliance and audit standards, particularly in the first few months of operation.

## 21 Tasks

### 21.1 Project Planning

Key tasks in project planning include:

- Defining the scope of the system and establishing clear, achievable goals for each phase of development.
- Setting a timeline for system rollout, including key milestones like system design, development, testing, and deployment.

### 21.2 Planning of the Development Phases

Development should proceed in phases:

- **Phase 1:** Initial research and requirements gathering from MES staff and student groups.
- **Phase 2:** Core system design, focusing on the backend ledger management and reimbursement workflows. This phase would be the most intensive.
- **Phase 3:** UI/UX design to ensure the platform is user-friendly and accessible to all MES members.
- **Phase 4:** System testing and debugging, including user feedback to refine functionality.
- **Phase 5:** Final deployment and training for users.

## 22 Migration to the New Product

### 22.1 Requirements for Migration to the New Product

To successfully migrate to the new financial platform, MES will need to:

- Train existing staff and student group representatives on how to use the new system.
- Ensure all historical financial data is correctly transferred to the new system and remains accessible for auditing purposes.

## **22.2 Data That Has to be Modified or Translated for the New System**

The data to be migrated includes:

- Historical financial data such as past reimbursements, transactions, and budget allocations.
- Any audit-related data to ensure compliance with financial reporting requirements.

## **23 Costs**

While MES already has existing infrastructure on DigitalOcean, additional costs may arise from:

- Server upgrades to accommodate increased resource demands from the new platform.
- Potential third-party services for payment processing or APIs (e.g., Plaid, Stripe).

## **24 User Documentation and Training**

### **24.1 User Documentation Requirements**

User documentation must:

- Be clear, concise, and accessible to both administrators and student group representatives.
- Include examples for common tasks, such as submitting reimbursements and tracking group budgets.
- Be easy to navigate, with a search feature and organized sections for quick reference.

## 24.2 Training Requirements

Training will involve:

- "Getting Started" guides for both administrators and general users.
- Video tutorials or live training sessions to walk users through the system's core functions.
- Ongoing support through help desks, FAQs, and troubleshooting documentation.

## 25 Waiting Room

The following features are under consideration for future development but are not part of the immediate project scope:

- SMS notifications to update users on the status of reimbursement requests.
- Authentication via McMaster CAS 2FA (Central Authentication Service) for enhanced security.
- Integrated Phone Application

## 26 Ideas for Solution

The proposed solution involves leveraging a form of the traditional web full stack, potentially (MongoDB, Express, NextJs, React) to handle both front-end and backend requirements. This stack allows for:

- Efficient data integration with existing MES systems.
- A scalable, modular design that can be extended with additional features like payment APIs or compliance tools.
- Maintainability after the capstone is complete



## Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What went well while writing this deliverable?

**Jacob:** Writing the requirements went well as requirements have been discussed and practiced in several prior courses, such as 3A04 and 3RA3

**Christian:** Sections 1-6 required a lot of information that had already been discussed in prior meetings, so their inclusion was more of a formality.

**Evan:** Splitting up the work was once again an easy process for this deliverable. Everyone was present for our meetings and active in our group chat, making it easy to communicate and stay on the same page.

2. What pain points did you experience during this deliverable, and how did you resolve them?

**Christian:** Some parts of Section 3 (Mandated Constraints) I was unsure if I had covered all bases of what constraints we would work under, and opted to mention only the constraints that felt imposed on us by the situation and stay away from potentially detailing more requirements rather than constraints.

**Jacob:** The main pain points were from NFRs where I did not understand the definition. I solved it by looking at old lecture slides that explained what each non-functional requirement entailed. 13.4 was also a pain point as "productization" is not a word, so I was unable to generate any requirements, but I just made a note of that.

**Evan:** The biggest pain point of this deliverable was how close the due date was to our TA meeting. Thankfully, the deadline was extended but we got a lot of feedback from our TA meeting that we would have had to implement within two days. For future deliverables, spacing the TA meeting to the deliverable deadline might help students to implement the TA feedback.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

Several requirements relating to security and scalability were generated or inspired by talking to Luke. However, the functional requirements would be more heavily drawn from speaking with stakeholders, as the focus during these discussions was functionality of the system rather than its characteristics.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project?

Our engineering design courses, as well as requirements and design courses such as 3XB3, 3RA3, and 2AA4 will help us be successful.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

**Christian:** The ability to culminate domain knowledge from different aspects of software development that we've been taught throughout courses into a full-fledged software product for a client will be crucial to the flow of the project. We have all learned about databases, cybersecurity, object-oriented programming, data structures, etc., and have even developed outside of school. However, the ability to bring this knowledge together in a cohesive and intuitive way will avoid countless problems during development concerning the integration of multiple technologies together.

**Austin:** A mindset: being able to overcome the anxiety that can arise from approaching a problem that you have never seen before especially at this scale. For most team members this is the largest scale project to have be worked on. This can obviously be daunting. A message to those that feel this way, just dive head first, learn and fail. Each failure is a stepping stone to success, do not be afraid to fail.

**Jacob:** For most of us this will be our first experience building a software system of this scale, so fullstack development knowledge will be necessary in order to succeed in this project.

**Evan:** Software development knowledge will obviously be essential to completing this project. Specifically, we will likely need to acquire knowledge of security principles and best practices. This project has us working with sensitive information so ensuring compliance with privacy regulations will take some research.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

**Christian:** Two approaches to familiarizing ourselves with full stack development could be researching existing software systems and how their design and structure work together. Another approach could be taking the technology one is most familiar with and learning how it integrates with other technologies, to start from a place of familiarity.

**Austin:** Well to reiterate again, I think this choice of "skill" is widely applicable to many areas of life, the capstone provides a safe mechanism for learning this. An approach to acquiring the skill of facing a challenge is to embrace the mindset of continuous failure towards eventual success. (*A bit unorthodox of a response, but I still think is equally valuable*)

**Jacob:** One approach to gain fullstack development knowledge would be to gain experience by practicing small-scale development or expanding on prior projects. Another way would be by watching tutorials on the internet or reading textbooks.

**Evan:** Two approaches to familiarizing ourselves with implementing security principles can involve pursuing online training or in-person workshops. Online training can involve online courses, YouTube videos or any other online certifications that teach about cybersecurity. The benefit of in-person workshops or hackathons is the ability to network and learn hands-on since it would be in-person.