Team Contributions: POC
McMaster Engineering Society Custom Financial Expense Reporting Platform

Team #12, Reimbursement Rangers
Adam Podolak
Evan Sturmey
Christian Petricca
Austin Bennett
Jacob Kish

This document summarizes the contributions of each team member up to the POC Demo. The time period of interest is the time between the beginning of the term and the POC demo.

1 Demo Plans

We will be demonstrating the reimbursement request feature of our program. We have created a submittable form that allows users to input data that the MES requires for handling reimbursements. The data is then automatically updated in our database where it will be stored for record keeping. We also have created a simple mock checkout page for handling transactions.

2 Team Meeting Attendance

Student	**Meetings**
Total	5
Adam Podolak	5
Evan Sturmey	5
Austin Bennet	5
Christian Petricca	5
Jacob Kish	5

3 Supervisor/Stakeholder Meeting Attendance

Student	Meetings
Total	2
Adam Podolak	2
Evan Sturmey	2
Austin Bennet	2
Christian Petricca	2
Jacob Kish	2

4 Lecture Attendance

Student	Lectures
Total	11
Adam Podolak	9
Evan Sturmey	11
Austin Bennet	10
Christian Petricca	7
Jacob Kish	6

5 TA Document Discussion Attendance

Student	Lectures
Total	3
Adam Podolak	3
Evan Sturmey	3
Austin Bennet	3
Christian Petricca	3
Jacob Kish	3

6 Commits

Student	Commits	Percent
Total	97	100%
Adam Podolak	42	43%
Evan Sturmey	11	12%
Austin Bennett	24	25%
Christian Petricca	10	10%
Jacob Kish	10	10%

Note (Adam): I usually merge branches, commit the template files, and/or do small formatting at the end of deliverables which is why my commits are slightly higher than the others. These commits are usually small and frequent so it may inflate my numbers a bit.

7 Issue Tracker

Student	Authored (O+C)	Assigned (C only)
Adam Podolak	16	4
Evan Sturmey	2	4
Austin Bennett	8	4
Christian Petricca	0	4
Jacob Kish	0	4

8 CICD

For the reimbursement platform, CICD will streamline development by automating testing, integration, and deployment processes. GitHub Actions will serve as the foundation for this CICD pipeline. Whenever code changes are pushed or a pull request is made, GitHub Actions will trigger workflows to automatically run tests, build the application, and verify code quality.

Through this automated testing and validation, we can ensure new code does not introduce bugs, enhancing reliability. After successful tests, GitHub Actions can deploy updates to a staging environment, and eventually to production. This automated cycle allows the team to focus on feature development, reduces manual work, and ensures consistent and reliable deployments, resulting in a faster, more stable development workflow for the platform.