Team Contributions: Rev 0
Software Engineering

Team #12, Streamliners
Mahad Ahmed
Abyan Jaigirdar
Prerna Prabhu
Farhan Rahman
Ali Zia

This document summarizes the contributions of each team member for the Rev 0 Demo. The time period of interest is the time between the PoC demo and the Rev 0 demo; the contributions prior to the PoC are NOT included.

1 Demo Plans

Our Rev 0 demo will demonstrate the complete system workflow and integration of all major components. The demo will begin with user sign-up and login using MAC address authentication, followed by event selection and registration. The flow will then incorporate event table/bus sign-ups with seat selection, demonstrating how users register for events and reserve available seats, followed by transaction processing through the payment system. The demo will also highlight role-based access control by presenting different user and administrator views, including the administrator dashboard for managing events, viewing registrations, and monitoring system activity. This end-to-end demonstration will show how the system supports event registration, payments, and management within a unified platform.

2 Team Meeting Attendance

Student	Meetings
Total	8
Mahad Ahmed	8
Abyan Jaigirdar	8
Prerna Prabhu	8
Farhan Rahman	8
Ali Zia	8

The team arranges meetings almost weekly, to clarify objectives and align on priorities on major deliverables, with additional meetings scheduled as needed, typically in the days leading up to major deadlines. We have held a total of eight meetings in the time period of interest. Every team member has been able to attend these meetings, even if occasionally arriving slightly late or needing to leave early.

3 Supervisor/Stakeholder Meeting Attendance

Supervisor's Name: Luke Schuurman

Student	Meetings
Total	2
Abyan	2
Ali	2
Farhan	2
Mahad	2
Prerna	2

Supervisor meetings were arranged when objectives and priorities needed to be clarified and all members attended those meetings.

4 Lecture Attendance

[For each team member how many lectures have they attended over the time period of interest. This number should be determined from the lecture issues in the team's repo. You can find the number of lectures in the time period of interest by looking at the Google calendar for the capstone course. —SS]

[NOTE: There will be approximately 1 lecture between the POC and Rev0 demos —SS]

Student	Lectures
Total	Num
Name 1	Num
Name 2	Num
Name 3	Num
Name 4	Num
Name 5	Num

[If needed, an explanation for the lecture attendance can be provided here. —SS]

5 TA Document Discussion Attendance

[For each team member how many of the informal document discussion meetings with the TA were attended over the time period of interest. —SS]
TA's Name: [fill in this information]

Student	Lectures
Total	Num
Name 1	Num
Name 2	Num
Name 3	Num
Name 4	Num
Name 5	Num

[If needed, an explanation for the attendance can be provided here. —SS]

6 Commits

[For each team member how many commits to the main branch have been made over the time period of interest. The total is the total number of commits for the entire team since the beginning of the term. The percentage is the percentage of the total commits made by each team member. —SS]

Student	Commits	Percent
Total	Num	100%
Name 1	Num	%
Name 2	Num	%
Name 3	Num	%
Name 4	Num	%
Name 5	Num	%

[If needed, an explanation for the counts can be provided here. For instance, if a team member has more commits to unmerged branches, these numbers can be provided here. If multiple people contribute to a commit, git allows for multi-author commits. —SS]

7 Issue Tracker

[For each team member how many issues have they authored (including open and closed issues (O+C)) and how many have they been assigned (only counting closed issues (C only)) over the time period of interest. —SS]

Student	Authored (O+C)	Assigned (C only)
Name 1	Num	Num
Name 2	Num	Num
Name 3	Num	Num
Name 4	Num	Num
Name 5	Num	Num

[If needed, an explanation for the counts can be provided here. —SS]

8 CICD

Continuous Integration and Continuous Deployment (CI/CD) will be used to keep the project stable and easy to maintain as new features are added. The team will use GitHub Actions to automatically build, and test the system.

For **Continuous Integration (CI)**, every time a pull request is opened, the pipeline will automatically run linting, type checks, and unit tests. This helps catch issues early and ensures that the codebase stays clean and consistent. It will also build the project to make sure that new changes do not break existing functionality before they are merged.

For **Continuous Deployment (CD)**, after merging into the `dev` branch, a staging build will be automatically deployed for testing and feedback. When

changes are merged into the `main` branch, a production build will be deployed. This keeps updates smooth and reduces the chance of deployment errors.

GitHub Actions will also send build or test failure notifications to the team so that problems can be fixed quickly. Environment variables and API keys will be stored securely using GitHub's built-in secret management. Over time, the team may expand the pipeline to include integration or end-to-end tests, but for now the main goal is to automate building and testing to save time and improve reliability.

9 Team Charter Trigger Items

9.1 Summary of Triggers

The team charter established several quantified triggers to help maintain accountability and ensure consistent contribution. These include:

- **Attendance:** Members are expected to maintain a 100% attendance rate for all scheduled meetings unless an acceptable excuse (such as a health issue or family emergency) is provided.
- **GitHub Activity:** Each member must demonstrate consistent GitHub activity every week, with issues actively in progress or commits made for review.
- **Task Completion:** All assigned work must be completed and delivered on time according to deadlines set during weekly meetings.

9.2 Trigger Violations

So far, the team has not experienced any major violations of the triggers outlined in the charter. All members have remained communicative, met deadlines, and maintained consistent GitHub activity. Minor delays in individual tasks have occurred occasionally, but these were communicated early and resolved collaboratively without impacting the overall progress.

9.3 Plan to Address Violations

If future violations occur, the team will follow the three-step escalation process defined in the charter:

1. First incident: verbal reminder during team meeting.
2. Second incident: discussion with the TA to address underlying issues.
3. Third or repeated incidents: escalation to the course instructor.

If the team finds that the current triggers are too strict or unclear, they will be revised by team consensus. For example, attendance expectations may be adjusted for legitimate scheduling conflicts, or contribution tracking may be clarified to account for non-coding tasks such as documentation or research.

10 Additional Productivity Metrics

[If your team has additional metrics of productivity, please feel free to add them to this report. —SS]