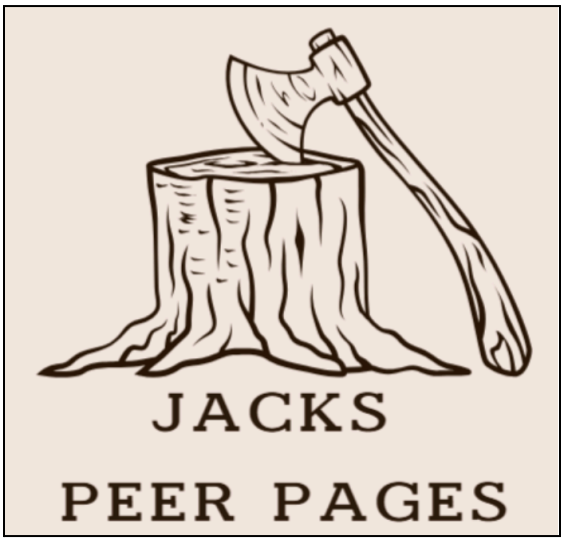
Team Standards



Jack's Peer Pages

September 26th, 2025

Sponsor: Dr. Keith Nowicki
Faculty Mentor: Md Nazmul Hossain

Team: Haley Berger, Haley Kloss, Jeremiah Lopez, Tyler Austin

Intro

The purpose of this document is to outline the expectations for group members throughout our Capstone project. It will include the team members, roles, a team meeting plan, standards for tools and documentation, and a structure for team self-reviews.

Team Members and Roles

- Haley Berger
    - Team leader, Coder, Release Manager
- Tyler Austin
    - Coder, SQL Architect/Database Manager
- Haley Kloss
    - Coder, Design Architect
- Jeremiah Lopez
    - Coder, Middleware Manager

Team Leader: Haley Berger will act as the team leader. She is responsible for handling communication with the mentor, submitting deliverables, and leading team meetings.

SQL Architect/Database Manager: Tyler Austin will act as the project SQL Architect. He will oversee the architectural decisions and take the lead on UML diagrams, as well as monitoring the follow-through of these plans during implementation.

Design Architect: Haley Kloss will act as the design architect. In this role, she will make figma prototypes for relevant features and web pages, then monitor their development to ensure that the product matches the client's expectations.

Release Manager: Haley Berger will act as the release manager. In this position, she will keep track of versioning and branching for the project. She will review and clean commit logs so that they are readable, accurate, and meaningful.

Middleware Manager: Jeremiah Lopez will act as the middleware manager. In this position he will test out the connectivity between the front end and backend of the code.

Team Meeting Expectations

Meeting Times: Thursdays, 2:30 pm weekly [In person, Engineering floor 2]

- Weekly team meetings are scheduled for Thursdays at 2:30 pm.
- Meetings will be held in the engineering building unless otherwise stated, in which case they will be held in the Discord #general voice channel.
- If a virtual meeting is scheduled or needed, members will be notified in advance via the Discord #general text channel.
- Impromptu meetings may be called as needed, with notification sent through Discord.

Agenda Structure: Each meeting will follow this general structure:

1. **Updates**: Each member provides a 2-minute update on progress since the previous meeting.
    - If a member cannot attend, they must submit a short written report (via Discord) at least 10 minutes before the meeting. Reports must include:
        - Completed work (bulleted).
        - Issues encountered (bulleted).
            - Potential solutions (if known).
        - Planned work for the next week.
2. **Review**: The team reviews upcoming deliverables, client/mentor expectations, and outstanding issues.
3. **Planning**: Tasks for the week are identified and distributed according to skill sets and availability.
    - Absentee reports will be reviewed, and the absent member will be notified of final assignments.
4. **Wrap-up**: Confirm individual responsibilities and deadlines before closing.

Minutes: Meeting minutes will be recorded using the shared Google Drive template. Minutes will be finalized within 24 hours of the meeting. Each set of minutes will include:

- Date, time, and format (virtual or in-person).
- Attendance.
- Summary of each member's update.
- Decisions made.
- Assigned tasks for the upcoming week.

Decision-Making Process: The primary goal is to reach a consensus or compromise with the team. However, in the case that this can not happen:

- The decision will be made through majority vote.
    - With four members: a 3/4 majority.
    - With three members (due to absence or abstention): 2/3 majority.
    - Or all other cases not listed.
- In case of an even split (e.g., 2–2), the mentor will only be contacted as a final resort. If the decision relates to design, both options will be presented to the project sponsor for

them to make the final decision. If a decision is still not reached, a coin toss will be enacted.
- For other major decisions, dissenting members' concerns will be documented in the meeting minutes.

Attendance: Attendance is required for each meeting.
- **Missing a meeting with notice**: Missing members must notify the team at least an hour in advance and submit a written report (see: Agenda Structure). Occasional absences are acceptable; however, if absences exceed 5 total or 3 consecutive weeks, the team lead will discuss with the member, and mentor involvement may follow if issues persist.
- **Missing a meeting without notice**: Immediate follow-up will occur (via email and text). Unless excused (e.g., emergency), the missed meeting will be recorded. Three total unexcused absences will trigger mentor involvement and possible disciplinary action.
- **Absence excusal**: In the case of an emergency, any absence resulting from an emergency may be excused and not counted as part of a total for any type of absence (that being, with notice or without notice). This will be pending based on team discussion and agreement or, depending on the circumstances, may be automatically granted (ex: medical).
- **Tardiness**: Members are expected to join on time. Consistent lateness (3+ consecutive times) will be addressed in a formal discussion.

Conduct:
- **Meetings will follow a general, consistent structure**:
    1. Opening statement from the team lead.
    2. Member updates.
    3. Discussion of current issues/deliverables.
    4. Confirmation of tasks and deadlines.
- **Expectations for constructive participation**:
    - All members are expected to listen actively, avoid interruptions, and engage respectfully.
    - If a discussion becomes unproductive, any member may call it out. The team lead will redirect focus.
    - If disruptions persist, the issue will escalate:
        1. formal team discussion
        2. mentor involvement
        3. meeting with the Capstone Organizer.
- **Interpersonal or design conflicts:**
    - Issues should be raised respectfully in meetings.
    - Members may not unilaterally change significant design elements without prior team discussion and agreement.
        - Testing anything is permitted, but it should not be pushed to production if it conflicts with the former agreed-upon design elements until agreements are revised and the change is permitted by team majority. This can occur through a Discord vote/discussion or a vote during a meeting.

    - Disputes will follow the decision-making process prior outlined in this document.

Tools and Document Standards

Expected Tools: The team will use a GitHub repository for all code development. The team is also expected to use VSCode, connected to the GitHub repo, for all development and PythonAnywhere for both testing and initial product production. Django is the framework for development, and SQL will be the database.

Version Control and Issue Tracking: The team will use GitHub as the central platform for version control, code review, and issue tracking.

- **Dependencies**: A requirements.txt file will be maintained to track project dependencies.
    - Each entry must include an explicit version number (e.g., package_name==1.0.0).
    - When a new library is added, it must be tested and documented in the requirements file before being merged into the main branch.
- **Commit Standards**: All commits must be:
    - Self-documenting: Code should be clear, readable, and maintainable.
    - Consistent: Follow agreed-upon naming conventions and style guidelines.
    - Well-organized: Functions and modules should be logically structured.
    - Commits must be made to a branch called production-ready rather than main. Code migrations from production-ready to main will happen on a regular release schedule to keep things consistent. Regular pulls from production-ready are expected of all team members.
    - Commit messages must be clear about what is being added.
        - Ex: '[Type]: short description
        - Possible types: fix, docs, refactor, test, chore.
    - ***Commits should be made regularly**, but not necessarily to main and production-ready. Each member should be prepared for a scenario where they may lose code on their local machine and have it ready in their GitHub branch.
- **Branching**: Branches should be clear. They should generally be structured such that:
    - *main*: Contains production code only.
    - *production-ready*: Holds tested, stable code that is prepared for scheduled production releases.
    - *dev_**: Used for feature or development work. All development branches must begin with dev- followed by a descriptive name (e.g., dev-login-auth) and/or the name of the person who is working on the branch.
    - **Hotfixes may be created directly from main** (branch name: *hotfix_**) and must be merged back into both main and production-ready once reviewed.
- **Code Review and Approval**:
    - Direct commits to main are NOT permitted.

    - All pull requests must be reviewed and approved by at least one other team member before merging.
    - If the code does not meet team standards, the reviewer will request changes. The contributor must revise and resubmit until approval is granted.
- **Issues and User Story**: All issues, bugs, and user stories will be tracked within the GitHub repository.
    - Each commit and pull request should reference the related issue number if applicable.
    - Completed issues will only be closed after the corresponding code is merged into the appropriate branch.
- **External Issues and Assignments**: Items external to the code production will be tracked in the task tracker spreadsheet. Coding items will also be tracked in the task tracker. The task tracker should be used as a collection of all tasks performed by the team.

Word Processing and Presentation: The team will use cloud-based tools to ensure collaboration and consistency across deliverables. All files will be stored in the shared Google Drive for easy access and version tracking. Final deliverables will be exported in the required format as needed.

- **Google Docs**: for all written documents.
- **Google Slides**: for presentations and reviews.
- **Figma**: for wireframes, UI/UX mockups, and graphical design.
- **Lucidcharts**: for all UML and related diagrams. Connects directly with Google Drive.

Composition and Review: For larger document deliverables, the team will divide sections among members, with one member designated as lead editor for each deliverable. The lead editor may rotate depending on the assignment. The process for deliverable development is:

- **Draft Submission**: Each contributor must submit a rough draft of their section to the editor at least 96 hours (4 days) before the due date.
    - This is to allow the editor time to make comments and request changes from each contributor. The editor should request changes at least 48 hours (2 days) before the assignment deadline.
- **Final Submission**: Revised sections must be submitted to the editor at least 24 hours (1 day) before the due date.
- **Integration**: The lead editor will assemble the final document, ensuring consistent flow, formatting, and level of detail.
- **Review**: Upon editor notification and if time permits before submission, each member of the team should review the final document. This is the responsibility of each member. The editor should notify the team when the final document is put together and inform them of when the submission will occur (at least a half-hour advance notice is expected).

Team Self Review

In addition to the class-level peer evaluations, our team will conduct internal self-reviews to encourage continuous improvement and accountability. These will take place after every major assignment submission as part of a regular team meeting. Each member will share a brief self-assessment that includes what they did well, what they struggled with, and what they can improve moving forward. After each self-assessment, the team will provide constructive feedback and work together to identify possible solutions to challenges raised. All feedback must remain respectful, focused on project success, and framed constructively. A brief record of the discussion will be kept in the meeting minutes so that recurring issues and progress over time can be tracked.