**Team 5 Product Backlog**

**Pleiades**

Cate Harrison, John Kim, Henry Rovnyak, Ethan Dawes

### Problem Statement

Students at Purdue and similar universities are beset by the plight of bad group members, poor coordination, and bad time management. Teams would benefit strongly from tools that streamline teamwork. Pleiades is a group work “toolbox” that provides these tools such as finding open time slots in group members’ schedules, assigning roles and responsibilities, and tracking progress of group and individual tasks. Pleiades provides a unified platform specifically designed for short term group collaboration.

### Background Information

One of the most commonly cited pain points of education is group work. Although group work is critical to growing communication, collaboration, and leadership skills, this process is often frustrating. All students, from middle school through to university, could benefit from a tool that makes communication and collaboration easier in group activities. We aim to do this by creating a set of tools to assign and review responsibility and accountability.

There exist some tools that accomplish parts of our goals. Trello and Github Projects are both products that allow you to create and assign tasks and organize them in a kanban board. While this may be enough for professional teams with established trust, they don’t provide the kind of detailed accountability or time tracking that might benefit smaller, short-term teams formed for school assignments. Pleiades will allow users to track how much time was spent by which users on which tasks, which increases accountability across members that may have just met. Shovel is a tool that allows users to plan and track time spent on tasks, but this is a single-user tool meant for organizing schedules, and is not a shared system. Pleiades will be a multi-user system that is designed for groups as the central unit.

### Requirements

### Functional

* As a user, I want to be able to
  + **Account**
    - Create an account with a email, real name, and password
    - Log in with a email and password so that I can see information for me and my team
    - (If time allows) login with social accounts (Google)
    - (If time allows) Reset my password so that I can recover my account
    - Upload a profile picture
  + **Group**
    - Create a group so that I can begin my project
    - Generate an invite link to the group so that I can collaborate with other students
    - (If time allows) Invite people directly in the app using their email
    - Click on invite links and automatically be added to that group
    - Be directed to the login page when clicking an invite link if not signed in and automatically being added to the group after signing in
    - Leave a group if I am leaving that group or changing group
    - Delete a group so that data can be deleted if wanted at the end of the project
    - See a list of groups I am in so that I can choose which group I want to view
    - View my teammates’ profiles, specifically names, profile pictures, and contact information
    - Create and view a group information panel with a project name, project description, and links in the description automatically converted to anchor tags
  + **Tasks**
    - View current tasks in a simplified 3 column kanban board
    - View who is assigned to each task
    - Create tasks so that I can track those tasks throughout the lifetime of the group project
    - Assign due dates to tasks
    - Assign time estimates to tasks
    - Delete tasks and mark tasks as completed
    - Assign/unassign tasks so that the entire team knows who is responsible for which tasks
    - Track time spent on tasks so that the team knows how much time has been spent by who on each task
    - (if time allows) View a personal window with all tasks I am assigned to across multiple groups
  + **Stats**
    - See a burndown chart so that I can track the team’s progress toward our goals
    - Split the burndown chart into a stacked line plot by user
    - Show the burndown chart by estimated time
    - See a pie chart with how much time each team member has recorded
  + **Calendar**
    - (if time allows) Upload my calendar from another program into Pleiades so that it can be used for finding team availabilities
    - Manually set availabilities if I don’t use a traditional calendar
    - View everyone’s availability as a calendar in a weekly view
    - View due dates of tasks in the calendar view in a monthly view
    - See a color coded view of the calendar/tasks indicative of the progress made
  + **Notifications**
    - (if time allows) View notifications in-app of updates and changes made to the project
    - (if time allows) Be able to change/customize my notification settings
  + **Voting**
    - Start a vote to decide project direction when consensus cannot be reached
    - Vote in a project vote
    - Display the results of of a vote

### Non-functional

* Architecture
  + Frontend written using the Svelte framework
  + Utilize components from shadcn-svelte
  + Backend written using Go
  + Use client side rendering to make the backend and frontend independent
  + Client-server communication will be done using websockets to allow real time updates
* Appearance
  + Use a minimalistic and modern style that is accessible and follows common, well known design patterns
* Security
  + Use a rigorously tested library to perform password hashing and comply with security standards (<https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html>)
  + Be cognizant to avoid and mitigate buffer overflows and XSS attacks
* Performance and Usability
  + Animations must be smooth and we must ensure that we hit 60 fps on mobile level hardware
  + Initial page loading times must be under 1 second including client side rendering
  + We will perform user studies to ensure that our UI design is intuitive and understandable
  + (if time allows) We will make our design responsive to mobile devices
  + (if time allows) Allow using the website offline and sync when back online
* Developer experience
  + 60% code coverage
* Have CI/CD so that deployments can be made quickly to the prod server
* John will host the deployment on his personal server
* We shall reify an impressive, useful product
* We shall get a good grade