

## **Team Working Agreements**

!Toohak

Toohakers

12 March 2024

### **Team Working Agreements:**

- Logistics
  - Work room: SNE for in-person meetings, Zoom for remote meetings with TA
  - Meeting times: Weekly stand-ups every Monday and Friday at 12:00pm; Weekly remote TA check-in meetings on Tuesday at 1:00pm
  - Project Repository: GitHub - <https://github.com/combinedrepo>
    - Location: Remote
    - Organization: Agile Scrum
  - Communication channels: Discord for daily communication, Trello for task management
- Development environment
  - Platform: Linux, macOS, Windows for desktop development
    - Virtual machines
  - IDE: Visual Studio Code for all development
  - Other tools: Git for version control, npm for dependency management
    - Frontend Frameworks: React for frontend web framework, Tailwind for CSS framework, Vercel for deployment
    - Backend Frameworks: Socket.io for real time communication, Hono for backend web framework, Digital Ocean for deployment
    - Databases: MongoDB
- Coding style/standards
  - Use camelCase for variable names
  - Use PascalCase for class names
- Work (Process) Patterns
  - Definition of Done: Code is considered done when it has been reviewed by at least one team member, has been integrated into the main branch, and has no known bugs
  - Team collaboration: Code reviews are mandatory for all merges into main branch
  - Collaboration with experts
  - Areas of responsibility: Assign team members to specific roles based on expertise; rotate responsibilities every sprint to ensure understanding
  - Work hand-off/integration: Use pull requests for code integration
- Product Design Patterns
  - UX/UI look and feel: Follow the color-scheme and keep the design simplistic and concise.
  - Product architecture:
  - Common approach to common problems:

## **Team Working Agreements**

!Toohak

Toohakers

12 March 2024

- Error handling: Print to console, multiple users UI test for bugs.

## **The Definition of Done**

DoD for user stories:

- Creating a quiz room
  - Users can create a room with a unique joinable link
- Joining a quiz room
  - Users can join a created room with a link
  - Users can view the other players in the room
- Playing the quiz with others with synced timings
  - Users can have the same timer as other players
  - Users will be given the question at the same time as the other players
  - Users will be shown the correct answer at the same time as the other players
- See if the answer I provided was right or wrong
  - Users will be told if their answer is correct or not after all players have given their answer or the timer has concluded
- View the top scores for the quiz
  - User will see a leaderboard at the end of the quiz showing the top 3 scores
  - User will be able to see if they were in the top 3 via a golden color indicating their name

**DoD for tasks:**

- Code checked into repository
- Code reviewed by team members before merging branches
- Manual tests pass

**Style Guide:**

- Code has no linter errors under the Google and Typescript ESLint configurations with the `npm run lint` command.
- Use camelCase for variable names
- Use PascalCase for class names