

Team Working Agreements

Product/Team name: Gym Essentials

Date: 7/21/2023

- **Logistics**

- Work room: Room 4372 in UCSC McHenry Library
- Meeting times: 3 Scrum Meetings/Week: Tuesday 4:30PM, Friday 4:35PM, Wednesday 2pm in McHenry Library 4372
- Project repository: <https://github.com/ShاونveerGill/Gym-Essentials/tree/main>
 - Organization: Sprint tasks are developed in their own branches and then merged into master
- Communication channels: Discord Server

- **Development Environment**

- Libraries and Frameworks: Javascript with React-Native framework
- Version Control Integration: Git
- Figma utilized for UI design approach.
- IDE: Visual Studio Code
- Development Machines: MacOS and Linux
- Other Tools: Expo

- **Coding Style/Standards**

- Organization: Model-View-Controller design pattern
- Style Guide: [Google Javascript Guide](#)

- **Work (Process) Patterns**

- Definition(s) of Done: [Definitions of Done](#)
- Team collaboration: Tasks that are assigned to more than one developer are done in VSCode live share and/or via peer programming.
- Collaboration with experts (SME: subject matter experts): N/A
- Areas of responsibility:
 - Each developer should be responsible for their assigned user story task
 - Also should be flexible with helping other team members as the project is a team effort
 - Provide consistent communication to the team on updates and issues
 - Should let Scrum Master know of any important concerns
 - Continuous improvement on skillset and time management
- Work hand-off/integration: Before hand-off, code must be tested and merged.

- **Product Design Patterns**

- UX/UI look and feel: Simple/Modern Black and White
- Product Increment Strategies: Layer by layer (layer 1: Backend/BD, layer 2: Controller, layer 3: Frontend/UI)
- Common approach to common problems

- Worked as a team and communicated merge conflicts issues when making pull requests
- Utilizing unit testing to make sure changes are tested before merged into master.
- Error handling:
 - The utilization of extensive try-catch cases especially in areas where data is being transferred between the database.
 - Making sure errors do not arise in the first place. Through unit testing each and every feature. Reduce points where errors can be made.