

Planetarium Team Agreements

Logistics

- Work room: BE2 302
- Meeting times:
 - Monday 12:15-1PM
 - BE2 302
 - Wednesday 6-6:15PM
 - Discord
 - Friday 6-6:15PM
 - Discord
- Repository
 - <https://github.com/Scrumbags115/plannertarium>
 - Main branch should always be a working product
 - Make new branches as needed for new tasks, then make pull requests to merge into main
 - Notify everyone before changing files outside of the lib directory
- Communication channels
 - Discord for general communication
 - Github for specific discussions about pull requests

Development Environment

- Platform
 - Everyone is on Windows 10
 - May try OSX VM to try developing for IOS using an emulator if needed
- Any IDE/editor works
 - VSCode
 - IntelliJ
 - Android Studio
- Flutter supports all platforms, so VMs etc are not necessary for coding

Coding style/standards

- Refer to style guide

Work patterns

- DoD
 - Task is done when:
 - Code pushed into PR and all requested team members have reviewed it (minimum 1)
 - Pull request was accepted into main branch
 - Documentation describing basic functionality (not too in depth)
 - Code compiles and runs without errors
 - General user story acceptance criteria:
 - All tasks are done
 - All tests pass
 - All acceptance criteria are met
- Team collaboration

- Pair programming if a team member feels unfamiliar with something and wants to work together
- Clear communication between team members who do different tasks in a user story
- Collaboration with experts
 - None of us are experts and none of us know experts, so we do not have any
- Areas of responsibility
 - We call any task that involves designing user interfaces a frontend task
 - Some team members will specialize in this so they can learn in more detail and work faster
 - We call any task that involves designing database calls or computations for functionality a backend task
 - Other team members will specialize in this so they can learn in more detail and work faster
- Work hand-off/integration
 - Until a user story is complete,

Product Design Patterns

- Look minimalistic, not too complicated
 - colors will be variables so we can change them later
 - not overwhelming to users
- Feel fast, responsive
- Architecture
 - DatabaseService class handles calls to Firebase for getting and setting information
 - Task class is an abstraction of all data related to a task, and handles creation, modification, etc
 - Event class is an abstraction of all data related to an event, and handles creation, modification, etc
 - User has daily, weekly, and monthly views for tasks and events
 - these will make calls and manage information with the Task, Event, and DatabaseService classes to display the correct information

Other Agreements

- Try to always include at least 1 piece of positive feedback when reviewing one another's work

Team Working **Agreements**



■ Logistics

- Work room
- Meeting times
- Project repository
 - Location
 - Organization
- Communication channels

■ Development environment

- Platform
 - Virtual machines
- IDE
- Other tools

■ Coding style/standards

■ Work (Process) Patterns

- Definition(s) of Done
- Team collaboration
- Collaboration with experts (SME: subject matter experts)
- Areas of responsibility
- Work hand-off/integration

■ Product Design Patterns

- UX/UI look and feel
- Product architecture
- Common approach to common problems
- Error handling
- ...

What working **agreements** does your team find useful?