

## Team Working Agreements

- Logistics
  - No meeting room due to Covid. We are meeting online 5 times a week on Zoom, and then we talk amongst ourselves on Discord, a chat client.
  - Meeting times:
    - Monday 4pm
    - Tuesday 2pm (TA Meeting)
    - Wednesday 3pm
    - Friday 4pm (TA Meeting)
  - Project repository
    - Location: Github. <https://github.com/AremanHashemi/HouseHub>
    - Organization: We are using XCode, as it automatically organizes projects.
  - Communication channels
    - Discord, a chat client.
    - Zoom, a video conferencing client.
- Development environment
  - Ide
    - XCode
    - In-Browser Swift development
  - Other tools
    - Github
    - Git
    - Cocoapods
    - Google Drive/Docs
- Coding style standards
  - <https://google.github.io/swift/>  
^ This is a guide on how to frame Swift code.

## Work Patterns

- Definitions of Done
  - When the feature/task is fully implemented, the developer who programmed it lets us know that he/she has finished the task. When a feature is functional, it is considered “done” when it matches the user story. “Did we build the right thing?” “Did we stick to the SCRUM framework?”
- Team collaboration
  - During our daily scrum meetings, we go over what we’ve completed and what needs to be done. We talk frequently on discord when we have issues and need help from each other.
- Collaboration with experts
  - So far, we have had meetings with the TA, but that is the only “expert” we have collaborated with. We have also used tutorials from youtube, as well as taken a look at the API of Swift and other tools that we are implementing in our project.
- Areas of responsibility

- We have split up, instead of front end or back end, into tasks with different specifications. Each person, based on their history, experience, and interests, signed up to finish certain tasks.
- Work hand-off/integration
  - We are able to split the tasks up between each person in a relatively straightforward manner. If someone had a complaint about their workload, we discussed alternatives

#### Product design patterns

- UI look and feel
  - We have a general idea of how we want the app to look as a whole. We have wireframes created so that we are on the same page about UI. Members go through and can set (i.e. background color) aspects in the app so it ties together
- Product architecture
  - App will be built in Swift and we will use Twilio as well as Firebase for backend aspects.
- Common approach to common problems
  - First step would be to contact team members to see if there is any solution any of us could offer. Next would be to contact the TA and see if that offers any solution. If both continue to fail, research more on a solution and then plan a different approach that does not involve the aspect that is giving problems
- Error handling
  - We have identified aspects that may cause some error. Between all of us, we can avoid developmental error, which comes from syntax or other minor details. The error handling we have not anticipated is runtime error as well as issues with mainly the backend. None of us have experience with Twilio as well as Firebase so we expect to have errors in that aspect as well.