

Communication methods

- Discord pairing stations:
 - Meetings
 - Voice/video & screenshare
 - Coding together
- Slack
 - Providing updates, absences, planning
 - Voting
 - Backup notification
 - File sharing
-

Standups

After general Junior/Senior stand-up, we'll get together on Discord in a Pairing Station. The first instance on the calendar when we have time allotted for our FEC project.

Be prepared to discuss:

- What I did since our last meeting
 - What I plan to do before our next meeting
 - What I am blocked on
-

Daily Retro

- 5 min meeting at the end of day
 - Talking points:
 - what worked,
 - what didn't
 - ask for guidance
-

How do we handle teammates late or missing meetings?

- Communicate our absences to the team via slack before hand and communicate when we're going to make up the time/ work later.
-

How are we going to not overwrite our team member's work, and managing tickets?

Possible Solutions:

- Live Server- one person share's their document through the liveshare link and we all work on the same document.
 - Git branches- Features are on a branch, and we do a push request when we're done.
-

How will you handle your git workflow and ensure that everyone commits to it? What standards and process will you use for code reviews and pull requests?

- All devs will have their assigned features/branches that they'll be working on, and will create pull requests *to the dev branch* in github when they are ready to commit their feature.
 - One dev will be identified to make pull requests to the main branch.
-

How will we communicate feedback with team members?

- We're using Github projects to manage workload and status of each dev's work.
-

How are we managing/assigning the work?

- Github projects.
-

Project management: assigning roles

David notes

- Just point all api routes to the same database
 - nginx is a load balancer, between the client and the server.
-

Kanban board

- Backlog
- In-Progress
- In-Review
- Waiting to merge
- Shipped
- Tickets- Always work from tickets. Reference your ticket ID or URL in your Pull Request. Ask team for feedback on your PR before merging it.
- Giving / getting feedback is a critical skill and should not be avoided for the sake of 'writing more features'

Testing your work

Unit testing is our responsibility and is a bare minimum with every pull requests. Aim to do more especially with end-to-end tests.

Aim for 20% of your time writing tests. Make it part of your code review process.