

- Responsible for sending recording to the database/rubric/marking?
 - 40-50 images, starting & stopping point
 - Look at timing
 - How long it takes them to name all the images on the screen
- Transcription of rapid naming?
 - Indicate when they start & stop
 - Baseline: how long they take to complete task
 - Extra milestone: transcribing
- User records: don't have user records (go for it if you want)
 - Instead of focusing on authentication, everytime someone loads a page, that instance is your user (keep it local for now)
- Advice: each session being an instance applies to tasks as well; can save this data for milestones
- UI design for tasks?
 - Suggestion: car moving across (for spelling task)
- Tutorial optional?
 - Kids might need to see an example the first time they do it
 - If optional, they might not know how to click on it
- Spelling: click and drag?
 - Kids can get frustrated by dragging (motor skills?)
 - Clicking might be enough
- UI
 - A challenge to build UI for young children
 - Tasks themselves look fine
 - Speaking from user perspective: should focus on what UI is going to be
- Rhyming/phonological awareness combined?
 - Should keep them separate
- Keep database/grading system minimal so far
 - Priority: interactivity
- How to place apples on the tree?
 - Default overlaying
 - Determines how interactive application is
 - What happens when you're developing on different screen sizes?

- Clicking on overlaid images is not reliable
 - Advice: look at dynamic canvases
 - Print image on interactive canvas

- Dashboard
 - Feedback: mouse hovering - child might not be using a device with a mouse
 - Not absolutely necessary for authentication - but can connect to currently implementation if we have our own authentication system (not a priority)

- Database
 - Schema looks fine
 - Concerned: what kind of database?
 - If we go down sql route: a lot of overhead work
 - Biggest problems:
 - Problems with sql databases in general
 - Require well-defined schema
 - Difficult to build applications
 - Advice: move away from sql completely
 - App will be hosted on a website for free
 - Postgres server: not free
 - Hosting becomes a problem
 - Bots might spam server's request and collapse the website
 - Sql is too complicated
 - Limits adding task (due to strictly defined schema)
 - Sql works better with backend interfaces
 - Sql is not json default; does not work well with javascript
 - Can get mongodb instead; can host 500gb of data and will be free
 - Provides everything
 - Works with javascript
 - No worrying about query statements/inserting

- Firebase
 - Builds in authentication, database, storage
 - Can start with 10 lines of code and have it set up (takes time)

- IPCA?
 - Confirm with eunice
 - From discussion: tasks are common, publicly available
 - Nothing confidential
 - Confidentiality comes from grading metrics (which we're not implementing)
 - No IP ownership of confidentiality on our part
 - (But confirm with eunice)