

We're all here, and we read the project description for our second deliverable. Met from 1-3 PM.

We should integrate layouts into the main branch on Github. Further work should branch off of the new main after we accept all the merge requests.

One way to do this, is use the main and dev branches. Main has good tested working code. Dev is what we're currently working on, so when you want to add something, branch from it, make a PR, push to dev. The main contains good tested stable code, whereas the dev is more advanced newer stuff which may not be as stable. If we're approaching a deadline and the main is fine and we want to try adding cool new features, we can add those into dev and not mess with the main branch.

We can assign a couple people to do the merge requests so there's some structure. If someone does the login page and it connects with the daily view, if we're pushing the login page, we want to have the people who connect to that page review the login page. We'll have a main reviewer, and probably also a second person to do a quick review to get an extra set of eyes.

We can maintain a list of features for things that should work, and to have a basic testing procedure so after each major change, we can go through and make sure everything still works well.

For merging our XML stuff right now, a manual thing done by one person is probably the best way.

The nav bar. There needs to be common code across the screens so when we click on it, it'll launch an activity.

Nick does a screen share and shows us how he's using the bottom nav bar. Currently he has it so the buttons open up fragments and not activities. This means he isn't duplicating the bar for the fragments. We only have it launch activities if there's a screen without the bottom nav bar. Paola says that we can start activities from fragments, so we could switch to that as well. We could try using fragments first and switch to activities if there's issues. There may be some code changes if we want to launch fragments within fragments and could be more work.

Aparna and Jie will work on merging the XML stuff manually into dev.

Things are kind of complicated. The main screen/activity is split up into two boxes/fragment holders. The bottom box is the nav bar and the top bar is the rest of the fragments. We're discussing whether all screens will show the nav bar or not. The moment we get rid of the nav bar, we should switch to a new whole activity for the screen, or hide the nav bar but hiding it is a bit messy. HabitActivity has the nav bar.

Let's decide which screens need the navbar.

We'll have two activities so far. The login screen, and then the main activity. The app launches into the login screen which then launches the main activity.

We've decided to go with fragments as much as possible.

What do we need to get done before the next meeting? The TA probably isn't expecting a ton yet. We've discussed and figured out how to organize our stuff which is important. We'll merge the screens based on this discussion. This is top priority and needs to be done ASAP. It's the foundation for the rest of our stuff. Nick will support Aparna and Jie on this task. The task is done when the main lists, viewing and editing habits, and possibly profile are linked up.

By the deadline, we need server communication. This means the frontend needs to connect to the backend (server).

Everyone should be creating issues on Github for themselves, and make sure you add it to the backlog! Once something is done, clear it from the backlog so it won't back up and clog.

How do we coordinate the UI and stuff?

Nick can do the main activity and navigation. Maybe make it a class. Aparna and I can each do two other fragments, converting activities to fragments or something.

Jie can do the home. It has a milestone history button which brings you to the event list that John made. Also Jie will do the profile.

John does the list and navigation backend to link them.

Paola will do the CRC stuff.

John can start with a UML and work with Paola.

We should maintain a record of who's doing what for each week, and deadlines. We could have this in Github's Projects, but a text file in Github's probably easier.

Nick started a wiki page to keep track of this.