

General tasks to get done this week:

- **IMPORTANT:** rebase Navi's branch with master ASAP – there are model changes that the branch will need to work, otherwise it won't run properly. * see note
- (#176) Hide edit/delete buttons for non-privileged users, using the IFPERM tag (see lines 84-96 of server/HtmlRenderer for documentation [don't edit this file], and constants/P for permission names) (Ainur)
- (#154) Update “new booking” page to week-view only (Navi)
- (#191) Create AJAX route for unavailable periods for booking page (Kristin, + Ainur to help if needed) ** see note
- (#155) Display unavailable periods on booking page (Navi + George + Fares) ** see note
- (#192) Update the Staff “isAvailable” method to accommodate the new booking model (Kristin)
- (#157) Finish full implementation of all password-related features, keeping the code as neat and DRY as possible (make use of the utils classes, if you haven't done so already; you'll need to check implementation in CRUD route for staff creation and CRUD route for change password) (Mus)
- (#193) Unit tests for all password/staff features (permissions, password changes, etc. – some of this is already done, afaik) (Mus + Danial)
- (#151) Change password reminders (Fares) *** see note
- (#123 & #124) Page title and loading indicator in the web view (Mus or Danial, whoever is less busy/wants these jobs)
- Liaise with the client to arrange a demo meeting some time next week (Kristin or Fares)

* Rebasing:

1. Commit everything that's edited
2. git checkout master && git pull
3. git checkout NAVIS-BRANCH
4. git rebase master
5. At this point you will most probably get merge conflicts. Should be easy to sort out, and the error messages tell you where the problem is. The easiest way to fix them is to open up IntelliJ and look at the files that are changed – go into each one, fix the conflict, then run...
6. git rebase --continue

** Things to do on Tuesday for this task:

1. Navi, George, Kristin and Fares to agree on a JSON data format for querying unavailable periods. The endpoint should take *in* the therapy ID and start/end dates, and give *out* a list of slots that are unavailable in that period.
2. Kristin to create a “dummy” version of the endpoint that returns some static data, in the above format, for any week. This will allow points 3 and 4 to happen at the same time:
3. George, Navi and Fares to start work on sending an AJAX query to the endpoint for each week in the calendar (as it is displayed), then displaying the returned unavailable periods on the calendar.
4. Kristin to start work on the actual endpoint, that will take in the details above and return the desired output, based on camera availability and staff availability.

*** Notes for Fares:

- I've given you this job *and* the booking page work because I want George and Navi to take the lead on that job; you're just there to help join up the front/back end if needed, and to help with jQuery/AJAX if needed.
- For this, you'll probably need to do some work in the prepareToHandle() method (this is in routes/DefaultRoute) that checks how long since the password was changed and sets some fields that can then be picked up in _header.html to display the necessary warnings.
- When the 90 day limit is exceeded, they should be blocked from going anywhere except the profile page to change their PW. Again, you can probably implement this in the prepareToHandle() method, but feel free to do it elsewhere if you think that would be better.
- Store the warning periods (83 days/90 days) as constants in the constants/C file.