**Role of Secretary**

The secretary is responsible for taking notes during the meeting but also helps the chair with preparing and following up on action items. In addition, the secretary is responsible for communication with team members and the teaching assistants and for organizing documents. This means that the secretary is responsible for submitting documents to Brightspace on time.

**Tips for taking minutes**

* Prepare the meeting well, so you know the agenda items and objectives.
* Limit the minutes to what was discussed and decided in the meeting.
* Make a clear note of the actions: what the agreement entails, when the action must be carried out, and who is responsible.
* Elaborate on the notes and share with team members and/or non-team members.

Take minutes of what was decided and what actions were taken

Record (so write down) what was decided and what actions there are. You are free to take detailed notes of the entire meeting. But it can take a lot of time and is not always necessary. Therefore, it is recommended to limit yourself to the main discussed topics and to list decisions and action points. This is sufficient for keeping track of your project meetings. You can, for example, use one list that is attached at each meeting with the new decisions and/or actions. You can think of tools like Miro or Trello and such to keep track. Each decision must be accompanied by the date (i.e., during which the meeting) the decision was taken. We provide a solution for this in the template below.

Taking minutes

**Agenda for the team meeting [Project].**

Location: Drebbelweg, TU Delft

Date: 22/02/2022

Time: 15:45-16:30

Attendees:

Kuba Trzykowski

Reinier Schep

Vian Robotin

Mana Mahmoudi

Laimonas Lipinskas

Simona Lupșa

TA: Damla Ortaç

**Agenda items**

**Announcements**

Moved up the Thursday meeting to 13:00.

**Approval of the agenda**

Everything was fine

**Summary of action points**

*Give an overview of the division of tasks. What is expected from whom and when? Make an action list.*

|  |  |  |  |
| --- | --- | --- | --- |
| Action items | Who is responsible | Planned for | Course Deadline |
| Exchanging HCI feedback with other groups | Whole team | Tursday | Friday |
| Fix the code of conduct | Whole team | Thursday/Friday | Saturday |
| Finish up Information Literacy asignments | For each their own | For each their own | Sunday |
| Have atleast 1 working API endpoint | Whole team | Thursday | Friday |
| 3B team question. Specific might mean by providing a specific situation like my teammate is slacking. | Whole team | Thursday | Friday |
|  |  |  |  |
|  |  |  |  |

**Extra notes**

# HCI draft

*We should exchage with other groups. We give them ours, they give us theirs. Should give back a valid amount of feedback. Send who wrote what in the HCI draft. Can divide by chapters or this person wrote it, this person edited it. Everyone has to have had input on it. As long as we have a decent draft on Friday we can add stuff until Sunday. For the work division just ping Damla on mattermost with a doc or some messages of who did what.*

# Code of conduct

* *For the assignment description we are missing clear overviews of the tasks. Outsider readers wouldnt be able to follow it.*
* *Target or ambition levels. End goal described not only by grades. Should be something like GRADE + sentence on other goals.*
* *Product. We don’t have a clear specific list of deliverables. For example we should add that at the end we will deliver code of the project, an HCI documment that goes with the code and probably something more. Also we are missing some concrete standarts. Should add a more specific test part like what percentage of tests coverage do we want/expect.*
* *Planning. We should add concrete ways how we divide work. Like volunteering. Also we don’t have a concrete way of setting deadlines. Could do something like atleast one day before the official deadline we review stuff. Issues should also be mentioned for how long shall it take for you to work on it. May add estimated hours in the issues.*
* *Behaviour. All members have to have had worked on both the back-end and front-end by the end of the project. For example until week 7 half of the people work on front-end, half on back-end. Or just rotate every week. Should add more concrete disagreement handling options. Like scenarios and solutions. For example: If 1 person disagrees do we ignore them or let them voice their opinion to try and convince everyone.*
* *Commitment. We don’t have any specific measures for quality. For the good chairman part could add a measure like there were not many extra questions orthat the agenda was good. For example in HCI lets say you assign the methodoligy. What qualifies that the work is not good/ not good enough.*
* *Meetings. We should add something like the planned date of Thursday meetings. Also add what are the goals of them(To work, to discuss). What if we want to cancel. Do we postpone or cancel or go through. How do we set it up if one person wants a meeting and asks for a last minute one. How many days in advance should we be informed on them.*
* *Decision making. If we do not reach consensus should we go with the decision with most votes?*
* *Concequences. What happens if somebody doesn’t do their work. Same as problems but with more examples. Like give another chance. Or help the person if he messes up for a deadline or something along those lines.*
* *Pass is 20, we have 12 points. Although we have multiple tries on it, it’s best to finish it this week.*

# API endpoints

*We are expected to use REST APIs. REST should be enough to implement everything. Can be anything like make a list of three questions and select one of them randomly. (Get, Put, Post, Delete)*

# Good gitlab practices and general tips

* *Allocate maximum one or two people at a time on an issue.  
  (Optional) If you have finished an issue and it turned out to be larger than expected document what extra you have added.*
* *Issues should have weights or due dates. That may help with work division. If you finished it in12 hours instead of 3 update it one gitlab. Should try to use Time estimations on Gitlab. In the descriptions of the issues it’s possible to do /estimate /spend and other fun stuff.*
* *Update the board. It’s a good way of tracking which issue is in progress. Could have lists like To do this week, in progress, in review, done. For the review part the assigned people should try to review in something along the lines of max 3 days. Issues should only be code related stuff.*
* *The project pipeline should always be green. This is done by always pulling dev, dealing with the merge conflicts on your end and only then pushing to dev.*
* *Try to avoid commiting like every 1000 lines of code. Try to implement in smaller parts and push as ocasionally as possible. Keep as compact as possible.*

# Random notes

* *A merge request can be deleted by Damla if we name it to be deleted.*
* *There is an all-seeing table for overall activity so should it may be a good idea to try to split commits, approval, comments and so on for it to be pretty equal.*
* *Try to comment more. For example what may be unoptimized in the code to be merged or I would’ve used a different data structure if everything is fine.*