# Teamwork

Contents:

Roles and responsibilities

We used a Model - View - Controller pattern/framework, and this was broadly how the tasks between our team were split.

Generally (but not entirely) the responsibilites were as below:

- Hendrik: the Controller, writing unit tests (with Justin), developing the CLI (with some input from myself) and early on was unofficially project manager.
- Justin: the Model, writing unit tests (with Hendrik), developing the database and entering in dummy data.
- Me: the View (with input from Justin), blackbox testing, some contributions to the CLI, taking the lead on the system seminar demonstration, collating and submitting both assignments.

Collaboration tools:

- Trello - we started using this but did not continue.
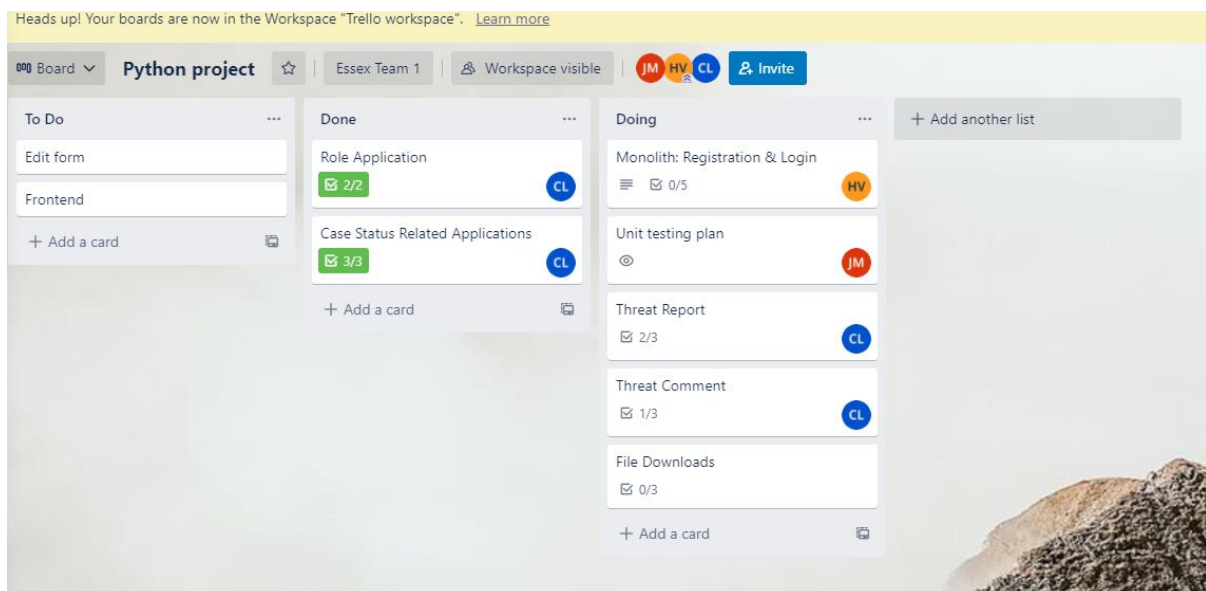
We found that we were collaborating on Discord effectively, and that using Trello caused extra workload.

- Discord (as mentioned below).
- Google Docs: Hendrik set up collaborative documents for the design proposal and report.
- GitHub: we collaborated on our code via each of us having a branch on our application's Github repository. We agreed as a team that we were to merge the main into our own branch before starting each coding session. We wrote and tested our own code in our own

branch before merging it back into the main branch. There were a few times where we had to go through *History* and revert to a previous version. We had a (mostly) agile approach, involving short sprints, this meant that we never had to revert far.

Meeting notes:

Our meetings were generally each Sunday at 13:00 UK time (on two occasions we also met midweek).

Most of our teamworking was through real-time or asynchronous Discord chats in between meetings. For example:

Jonny Mason 16/09/2021
Hi everyone, here's my dashboard. I've worked out a way to truncate the description for each case so once it gets beyond a certain width it cuts off and ends with "..." . i was thinking this is probably better than including the whole description on the dashboard

Dashboard.zip
14.99 KB

Jonny Mason 18/09/2021
Hi everyone, I've added some extra text into the document. I've written it in red to show what I've added. I haven't made a developer view prototype as I think I need some ideas on how this will look. In my mind it partially overlaps with the administrator's role.

Jonny Mason 19/09/2021
Thanks Justin 👍 . Am I uploading the mock-up files or just the activity diagrams?
I've had a read through and just changed one part from first person to third person. I also found out that Police logo isn't under creative commons license, so added small bit of text in the appendix
Essentially it seems fine to use it unless we upload it to our eportfolios
I'm off out for a bit, but am thinking of sending it tonight. Maybe 22:00 UK time. Is everyone ok with that?

Jonny Mason 06/10/2021
Hi everyone, I've made some changes to the read.me file as a few things are different on Windows, for example to create a virtual environment is "venv\scripts\activate" (rather than "source venv/scripts/activate"). Also Python -m venv venv should be used on Windows (rather than specifying Python3) (I'm not sure if this changes if the user also has Python2 installed). I've done some testing on Linux Mint and everything in the read.me works perfectly (so I assume it works with all Debian based distros)

Jonny Mason 21/10/2021
do you think I should add more about how we used SQLAlchemy to implement the MVC framework with Flask?
I'm thinking of just adding this when we talk about the folders:
Models: we use SQLAlchemy to save/retrieve objects in our database
Controllers: contains the functions handling our routes
Views: our html files

👍 2

Jonny Mason 25/10/2021
Hi everyone, one slight issue. I've been through. I've been doing the blackbox testing based on the below file and for some reason commenting in the chat box doesn't work for Editors and Approvers (edited)

Could not render this file preview.

functionalTests.html 2 KB

I'll have a look tomorrow and see if I can work it out from Git history
Do you remember the last time it was working?

My notetaking from our regular Sunday meetings was poor, and this has caused an issue with lacking these to put in my e-portfolio. Most of the notes I took were on, now long-lost, pieces of paper. Those I made on Word are below.

12/09/2021

Decisions:

- Use Jira/Trello when developing
- Have the option to download the whole table, perhaps next to the plus button
- Need to add a dashboard
- Need to have admin option to disable and reactivate users
- Need to add comments box

10/10/2021

- Remember run init in CLI each time.
- Can create fake email addresses at temp mailio.com
- Debugger now working

My tasks:

- Add front-end to login page
- add bandit to -r requirements.txt

General reflections

I found developing in a small team highly beneficial for several reasons:

- Learning experiences: early on I was afraid to ask others questions and for help, as I was the least experienced at programming. However, having approachable and helpful colleagues helped me see that it was worth sometimes feeling ignorant to gain the benefit of their expertise. An example of this was understanding the Model-View-Controller pattern/framework. I had read about it and had a vague understanding from covering it academically, but I did not grasp how it would be implemented until having my team colleagues explain it to me in simple terms.
- Social support: this was a fast-paced module and I found working with others normalised the experience of struggling to balance work and study priorities.
- Motivation: feeling part of a team, and not wanting to disappoint others, was a great motivation to get my contributions done in timely and effective manner.
- Friendship and contacts: during the project we developed a friendship with each other and are now contacts on Linkedin and continue to be part of a Discord chat. We all come from quite diverse areas of technology and will keep in touch, and benefit from each other's expertise.
- I found a better IDE. I moved from using NotePad++ / Visual Code Studio to PyCharm because of a discussion between our team. I have found this a better experience, especially the superior automated refactoring, and will use this in future when writing Python code.
- I cannot think of any negatives. I feel I perhaps I learned more from team colleagues than they learned from me. However, I did relay expertise on product management, GDPR, WCAG 2.1, Bootstrap and UI/UX design.