Modern Application Production

Final Project Report

Renat Valeev

Marina Nikolaeva

Alfiya Mussabekova

1. Content

During this course we implemented a simple database management system which was designed to manage hospitals. The system should be able to keep track of the records of the doctors, patients, nurses, and other hospital staff. The main aim is to minimize the paperwork of the hospital as minimum as possible, if not completely.

2. Before/after

Initial state: started from scratch, we only had an idea of a final product

Initial backlog:

1. As a receptionist I want to find people by their names
2. As a receptionist I want to be able to manage patient's appointments
3. As a receptionist, I want to create patient history so that each patient will have one
4. As a doctor, I want to edit patient history so that I won't forget what happen to the patient
5. As a doctor, I want to see the upcoming appointments so that I can prepare for the patient
6. As a patient, I want to get the medical certificate so that I can get sick leave
7. As a patient, I want to setup/cancel an appointment with the needed doctor so that I can go for treatment
8. As a receptionist, I want to assign call to the ambulance so that it can provide fast emergency help
9. As a patient, I want to see the schedule of the doctors so that I can match my schedule with the doctor's schedule
10. As a user, I want to get notifications about the future appointments so that I remember about them
11. As a patient I want to be able to manage my appointments
12. As a receptionist, I want to be able to add new patients to the system so that I can manage them later
13. As a user, I want to go to the profile page so that I see account information and future appointments
14. As a receptionist, I want to manage information about doctors, patients, appointments so that I can reduce paperwork

Final state: GUI connected to PostgreSQL DB via Java with possibility to create new records inside, such as patients, their history and personal information, doctors, etc.

Final implemented backlog:

1. As a receptionist, I want to find people by their names
3. As a receptionist, I want to create patient history so that each patient will have one

12. As a receptionist, I want to be able to add new patients to the system so that I can manage them later

13. As a user, I want to go to the profile page so that I see account information and future appointments

3. Gantt / Schedule / Organization

During the sprints we created the main branch *dev* and three branches for each of the group members.

After each sprint we merged work that was done to *dev*. At the end of the project all changes were
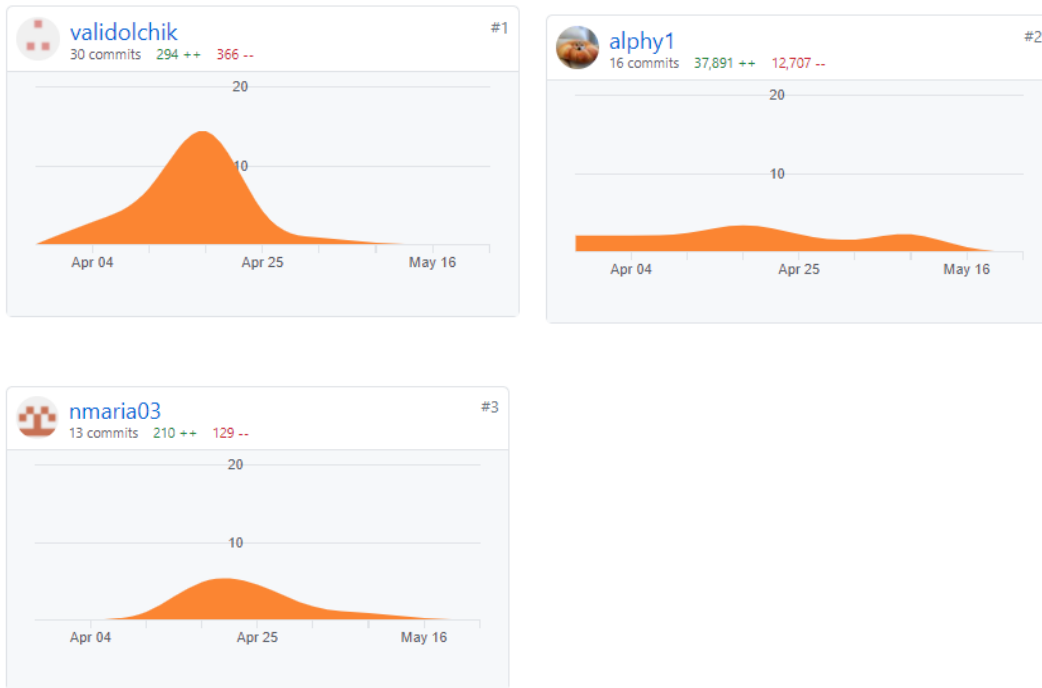
merged to master.

In total, we managed to go through 5 sprints, first of them was, in fact, Sprint 0 - preparation of initial

client backlog. Our team spent about 5-7 hours per sprint to complete the project.
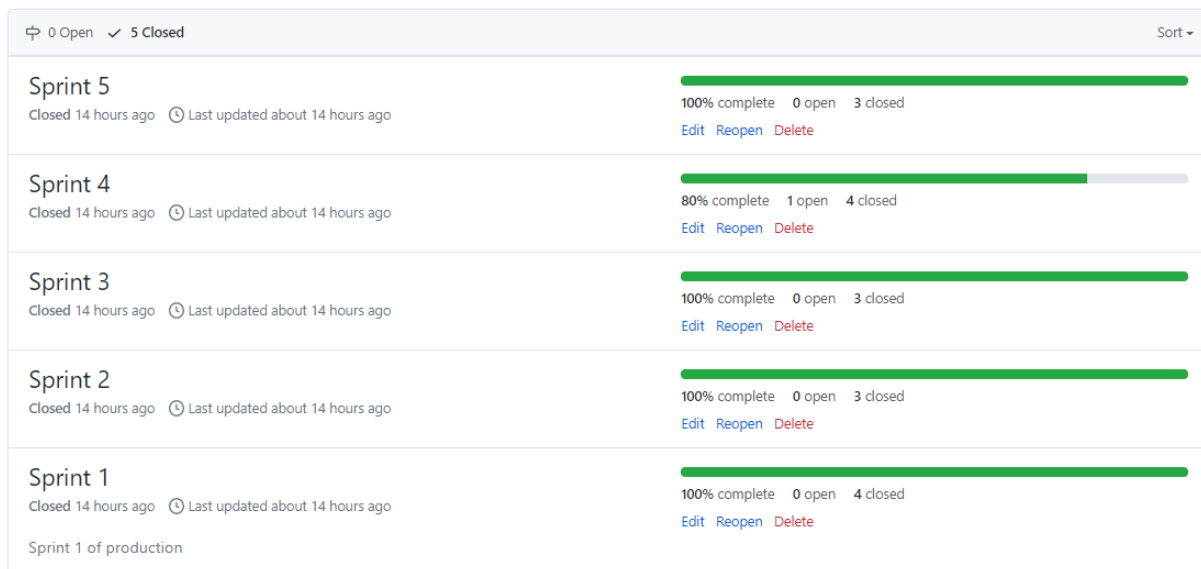
Here we may include our personal burn down charts(link):
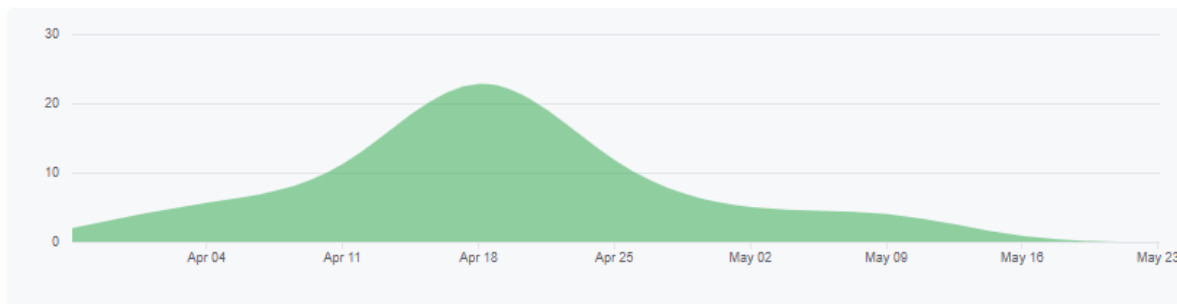
4. Traceability links demonstration

Assume you want to see why following code appeared, i.e. which issue and US this code connected to:

We can go to the History tab to see commit history on this particular file:

Then we can choose particular commit and see related information:

Here we can view changes presented by commit and linked pull request(circled with red):

Clicking on number near will open linked pull request and related information:

Scrolling a little will show linked issues:

Opening this issue will show all information about this issue, including to which US it is connected:

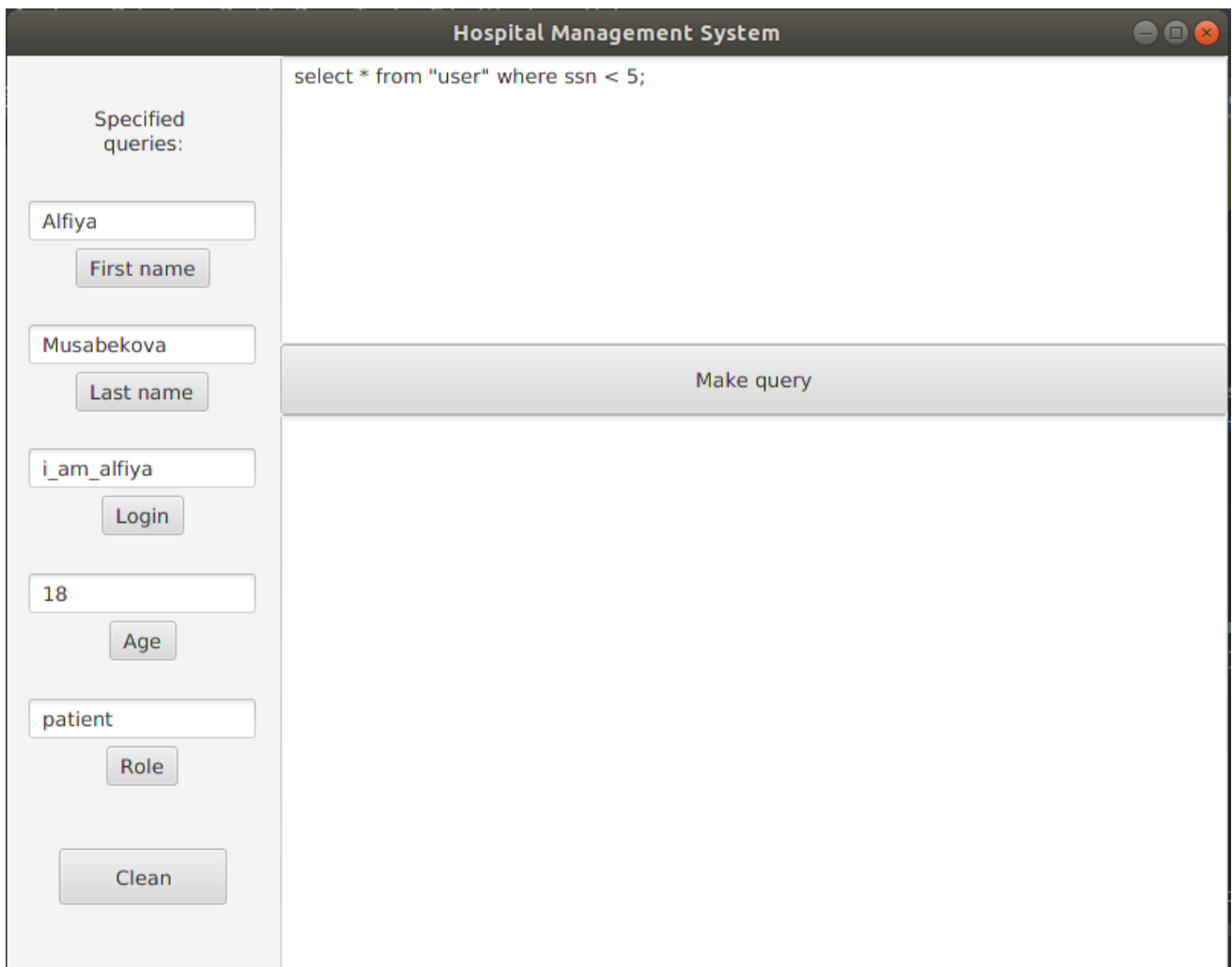In this case this issue connected to US #3:

With following content:

5. Documentation

Relation Scheme of database generated automatically and Use Case Diagram can be found in *Readme.md* on the main page of our github repository.

Instructions on how to set up an environment for development purposes are shown at the top of *Readme.md* on the main page of our github repository.

Our project is not fully completed for users, but this is how it will look at the end(link):

Users will be able to search people by their name/login/age or role. Also there is an option to create new queries for more advanced users. Not all of the fields have to be filled, search can be performed by one of the fields.

6. Tests

Unfortunately, we were not able to write tests.

7. Release

About CI: Managed to gain some information about how to work with Github Actions, write some workflows for it, but didn't manage to write a test, that is why not implemented as it should be.

About release, tags: Currently one release, available from the main page. There you can download it to local machine and continue developing

8. Quality assessment

We did not use quality assessment tools.

9. **Self-criticism**

Points for proudness:

1. Manage to implement better version of project, which have some pretty looking GUI, and able to handle some of the initial product stack

2. Teamwork - every sprint we gathered and shared with progress and have done during the sprint, discussed the next steps in development, helped each other

3. Get some knowledge on Github Actions and how to configure it, touched Maven

4. Learned how projects, such as ours, should be managed in order to have continuous progress on it.

What should have been done better:

1. Setup everything for development in the 0th sprint in order not to spend time on it during sprints.

2. In order to make some valuable product, more hours should be dedicated to it in order to be constantly aware of status and not spend time revising what needs to be done in the current sprint.

3. Tests should be implemented in early stages of projects, so that it is easier to add new.

4. Solid knowledge in CI and how to implement it for project

10. **References:**

   a. https://github.com/IU-MAP/project-ram

   b. https://jmbruel.github.io/InnopolisModernApplicationProduction/