SC-T-303-HUGB - Software Engineering

Sprint 4

Reykjavik University October 2, 2019

1 Workload

Due to recent comments/questions, I would like to repeat and explain a statement from the original project description: You are expected to only spend a limited time (I wrote 6.5 hours per week) on this project. It is your responsibility to organise your work in a way that this works out. This means proposing user stories to your PO that can be achieved in the limited time (with room for re-planning), and structuring the work in a way that allows you to work iteratively. It also means to separate tasks in such a way that the responsibilities in the team are clear and that you don't have to cover for a team mate that has not done his/her share. In turn, you can expect that we will take this into account during grading (which we have very much done already in the first two sprints).

2 Sprint 4 Introduction

In sprints 4 and 5, you will write a frontend (a graphical user interface) to another group's backend. That means that you abandon the work on your backend that you have done in sprints 1 to 3. Instead, you will have to understand another group's code and domain, and write software that corresponds to that interface. Sprint 4 is primarily a planning sprint, with little to no coding involved, while sprint 5 will be implementation-focused.

The backends will be assigned to groups on Thursday morning (3rd October 2019). The original groups that produced the backends will not be required to explain or support 'their' backend anyhow. Instead, we will create specific folders on Piazza to discuss the backends among the groups (and the TAs) with the same backend. It is explicitly allowed to explain functionality to each other, as long as artefacts for this sprint are not shared (UML diagrams, examples for a group's comparison to their own work).

3 Planning Sprint 4

1. **US1:** As a Product Owner, I want to know who is the group's ScrumMaster, so that I have a clear point of contact.
2. **US2:** As a group, we want to have a clear task list, so that we can see progress and identify who is responsible for what.

In this sprint, we will skip the user stories, since the sprint is focused on understanding the backend you have been assigned. However, we require you to maintain a task list (US2) throughout the sprint, so that it's clear who has been assigned what concrete task(s). This task list will evolve throughout the sprint.

4 Running Sprint 4

1. **US3:** As a Product Owner, I want to be able to see that the group has acquired an understanding of the assigned backend, so that they can propose meaningful user stories to implement in Sprint 5.
2. **US4:** As a Team Member, I want to spend sufficient time on learning PySimpleGUI, so that I can be productive in Sprint 5.
3. **US5:** As a Product Owner, I want to be able to get a quick overview of the standup meetings and important decisions, so that I can estimate the progress of the group.

The main task, addressing US3, will take most of the time in this sprint. Familiarise yourselves with the project description, the user stories, and the interface code of the backend you have been assigned. If needed, look into the remaining code as well. We expect three concrete outcomes for this US:

1. A UML component diagram displaying the backend component (as a black box) together with its interface.
2. Several UML sequence diagrams displaying the interaction necessary to fulfil (the success case of) the backend's user stories. See a more detailed explanation in Section 5.
3. A short (less than one page, 12pt font size) discussion of the differences in your backend and the backend you have been assigned. Discuss the technical differences, not the differences in the domain (e.g., do not write things similar to "We did a car rental application, they are doing book rental"). For instance, the organisation of the code might differ, the way and order how things are called might differ, etc. We expect a professional tone and will not accept any kind of belittling.

Apart from US3, reserve sufficient time (e.g., 4 hours per person) to familiarise yourself with PySimpleGUI (addressing US4). The official documentation (`https://pysimplegui.readthedocs.io/en/latest/`) gives numerous examples of how PySimpleGUI can be used to build everything from basic to advanced graphical UIs. The focus in Sprint 5 will be on the implemented functionality, not the UX/look and feel - so spend your time on figuring out how to do things, not on the design part. Try, as a group to think about how to structure the user interface in a way that you can work on user stories as independently as possible.

Throughout the sprint, keep doing standup meetings and discuss in your group. We will continue the retrospective format used after Sprints 2 and 3. Keep your decision protocol and summaries of the standup meetings (US5).

Overall, the intention of this sprint is to prepare you for (a) being able to propose sensible user stories and (b) directly start programming in Sprint 5.

5 Sequence Diagram Format

(UML) Sequence diagrams can be drawn at varying levels of detail depending on their purpose. For this project, the focus lies in communication: We want that you as a group understand how the backend functions and how this connects to the user stories. Each diagram should therefore depict the interaction for a single user story (the success case only). You do not have to draw diagrams for all user stories covered by the backend. Instead, prioritise which user stories are most important and then draw the diagrams according to the time you have.

Since the backend can be treated as a black box, we do not need to know the details of the backend code, but just the interface. Therefore, design your diagrams in a way that they show the interaction between the frontend and backend (as simple black-box entities). As an example, consider the (very simple) user story where a system allows users to register using a user name and a password. The system then returns a confirmation, and no further requests are required to fulfil the user story. Figure 1 depicts a UML sequence diagram that visualises this interaction. What's going on inside "backend" and "frontend" is irrelevant to our purpose: A software engineer could now successfully implement the user story in the frontend (even though more detail on attribute types, exceptions, etc might be required.). Not that user stories might require considerably more than one function call.

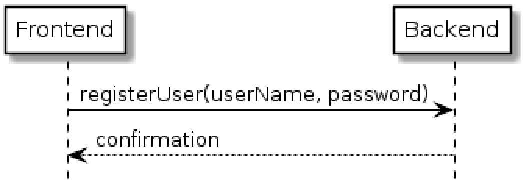

Figure 1: Example Sequence Diagram for Registering a User

Note that WebSockets use message exchange. This does not directly map to UML, since UML considers method (operation) calls. Therefore, you will have to devise a way to describe this in a way that is clear (e.g., by describing the message in a typical method format as done in Figure 1).

6 Technical Constraints

The following constraints apply to Sprint 4:

- UML syntax needs to be followed, but not strictly. For instance, specific arrow types are not important as long as the conveyed meaning is clear.
- Use your tool of choice for drawing diagrams, as long as they are readable.
- Keep using your existing repository. Do not include the code of the backend you have been assigned.

You will have to follow all of these constraints.

Submission and Assessment

Sprint 4 deadline: Wednesday, 16th October, 23:59
Make sure all relevant files are committed on time. Any late commits will be ignored.

For Sprint 4, make sure the following parts exist:

1. Decision protocol and summary of standup meetings in the docs/sprint4 subfolder
2. UML component diagram as a picture file (PNG, JPG, or PDF) in the docs/sprint4 subfolder
3. UML sequence diagrams as picture files (PNG, JPG, or PDF) in the docs/sprint4 subfolder
4. A comparison report (1 page, no cover page) as a PDF file in the docs/sprint4 subfolder

The individual part is submitted through Canvas (assignments are named by sprint) - as in Sprint 1.