University of Zurich[UZH]

HASEL

Software Praktikum (SoPra) - FS22

Milestone 4 - Assignment

1 General Information

Milestone 4 is the last Sprint of SoPra, concludes the implementation phase, and the Software Engineering Lab. The deadline is on Sunday, **29.5.2022 23:59 CET** and includes the deliverables listed below (exception: "Beta Testing"). On the following Monday, **30.05.2022**, we will have the final presentations to showcase your achievements.

1.1 Deliverables Overview

Report: as PDF with a name of form *FS22-Group-XX-M4-Report.pdf*.
Presentation Slides: as PDF with a name of form *FS22-Group-XX-M4-Slides.pdf*.
Beta Testing: of another groups application by **Sunday, 08.05.2022 23:59 CET**.
Source Code and Project Board hosted on GitHub (git tag "M4").
Final Application: hosted on Heroku.
Team Collaboration Reflection (weekly, individually): completed after your TA meetings.
Continuous Progress (weekly, individually): completion of 2 meaningful development tasks.
Brownie Points (individually): submitted via OLAT.

2 Assignment Description

The remaining weeks are about putting the finishing touches on the project, testing and stabilizing the application features, summarising the progress, and preparing the final project hand-over.

2.1 Beta Testing

Having worked continuously on implementation during M3, each group should now have a running prototype. Therefore, it is time to exchange feedback! In the first week of M4, another group will give feedback to your group. Similarly, you will have the chance to beta test another group's project. Schedule a 1-hour time slot with your team members in the first week of Milestone 4. The project your group is supposed to give feedback on can be found in the OLAT forum. Please take about 30 minutes to examine the other group's application thoroughly. In the remaining time, compose short, written feedback for the other group. Do not forget to include positive parts as well as suggestions for improvement. If you find bugs, make sure you provide enough details to reproduce them whenever possible. Feedback about the usability is highly appreciated, too: If you do not understand some parts of the application, let the developers know.

2.2 Finalizing the Project

In the last Sprint (M4), the goal is to finalize the last features of your project to comply with your specifications and the SoPra requirements. Further, you should focus on the stability and usability of your application. By the end of M4, everything should work without noticeable hiccups, and the interface should be user-friendly.

2.3 Testing

Proper software projects are extensively tested. Therefore, we expect you to have a high degree of test coverage for the server in terms of unit, integration, and REST interface tests. The whole test suite must reach at least 75% coverage. Check SonarQube for test metrics. We expect that every development task in the back-end has at least one good test. Further, every REST endpoint needs testing.

2.4 Summarize the Project with a README

Once you are wrapping up the development, meaning the majority of the features are fully implemented and working, the usability of your application is good, and the tests are passing, it is time to document your work for others with a README [1].

In both your GitHub repositories, create `README.md` files and write about the following aspects using the Markdown notation[2]:

- **Introduction:** the project's goal and motivation.
- **Technologies** used (short).
- **High-level components**: Identify your project's 3-5 main components. What is their role? How are they correlated? Reference the main class, file, or function in the README text with a link.
- **Launch & Deployment**: Write down the steps a new developer joining your team would have to take to get started with your application. What commands are required to build and run your project locally? How can they run the tests? Do you have external dependencies or a database that needs to be running? How can they do releases?
- **Illustrations**: In your client repository, briefly describe and illustrate the main user flow(s) of your interface. How does it work (without going into too much detail)? Feel free to include a few screenshots of your application.
- **Roadmap**: The top 2-3 features that new developers who want to contribute to your project could add.
- **Authors and acknowledgment**.
- **License**: Say how your project is licensed (see License guide[3]).

You can find a good template[4], as well as a curated list of good READMEs[5] online.

[1] `https://www.makeareadme.com`
[2] `https://guides.github.com/features/mastering-markdown/`
[3] `https://choosealicense.com/`
[4] e.g., `https://gist.github.com/PurpleBooth/109311bb0361f32d87a2`
[5] e.g., `https://github.com/matiassingers/awesome-readme`

2.5 Reflecting on Learnings & Challenges

In your report, reflect on challenges (technical, planning, coordination) you were facing in the last months. What did you learn? What would you do differently next time? What helped your team work well together? Compile the challenges to a short list (bullet points are enough).

2.6 Project Maintenance with GitHub

While working on the last features and the usability of your application, you are still required to ensure traceability of your progress. Your GitHub Issues and project boards should always be up to date. There are two ways to achieve consistency and traceability between your source code and your development tasks:

- **Manual:** Open a GitHub Issue and type the SHA[6] of the commit as an Issue comment. You can then proceed to close the GitHub Issue manually.
- **Automated:** Reference and close GitHub Issues in your git commit message[7].

Note: we expect every completed development task to link to at least one commit.

2.7 Weekly TA Meetings

For the remainder of the Milestone 4, you will keep having regular weekly Scrum meetings with your teaching assistant. During these short status update, every team member should provide a 3-minute update where they answer the following questions using the "Sprint Backlog View":

- What did I do last week?
- What will I do this week?
- What are the obstacles to progress?

3 Grading and Deliverables

In order to pass the Software Engineering Lab, you will need to pass Milestone 4. You will receive feedback on your deliverables, including an assessment (either pass or fail) in the upcoming days after the deadline.

Report
The report should be submitted as a PDF to OLAT with a name of form *FS22-Group-XX-M4-Report.pdf* by the group leader. Please make sure the title page contains the group name, group leader, and information about all group members (name and matriculation number). In the report, briefly introduce your project and its motivation/purpose, provide screenshots of your application (including captions), and compile the learnings and challenges as a bullet-point list.

Presentation (Slides)
Your final presentation should focus on your implementation achievements and take no longer than 4 minutes (hard cut-off). We suggest recapping your project idea, motivation, and major requirements before proceeding to showcase your application during a live demo. The demo's goal

[6] `https://docs.github.com/en/github/writing-on-github/autolinked-references-and-urls#commit-shas`

[7] `https://github.com/gitbucket/gitbucket/wiki/How-to-Close-Reference-issues-and-pull-request`

is to show that your application complies with the requirements and runs without bugs/problems. Notice that we expect the demo to be live (but take some backup screenshots if anything goes wrong). Further, we suggest you focus on the exciting parts of your application: feel free to skip the setup phase (e.g., registering users or setting up a lobby) to have more time to showcase the core parts of your project.

The slides should be submitted as PDF to OLAT with a name of form *FS22-Group-XX-M4-Slides.pdf* by the group leader. Finally, the title slide should consist of the group name and the names of all group members. Underline the name of the presenter(s) on the title slide. The slides and the presentations have to be in English. Each team member has to present at least once (M1+M2, M3, or M4).

Beta Testing
Please submit your feedback as PDF to `sopra@lists.ifi.uzh.ch` **before Sunday, 08.05.2022 23:59 CET** so we can forward it to the other group. It can be a bullet-point list and should be around 0.5-1 A4 pages.

Source Code and Project Board
The source code on GitHub is submitted implicitely by adding git tags "M4" to commits in the "main" branches that should be taken into consideration for grading. Similarly, your GitHub Projects board is submitted implicitly. The board should show your Product Backlog, including user stories and development tasks, and your finished Sprint Backlogs from M3 and M4. We will assess the overall quality and consistency of your project management activities.

In terms of testing, we will check whether your tests are meaningful, i.e., they serve the purpose of identifying bugs. Tests that cover getter/setter are not valuable. Aim for tests that deal with the core logic of your application. We expect that every task (from the back-end) has at least 1 test associated with it and that you achieve a test coverage of 75% on SonarCube.

Final Application
By the end of Milestone 4, your application should fulfill its primary purpose, comply with all SoPra requirements of the first assignment sheet, and run smoothly on Heroku. We will mainly assess your project's functionality, completeness, robustness, and usability. The deployed application should run smoothly and not have more than one major problem or 4 minor problems. A major problem would be the crash of the application or rendering issues inhibiting users from interacting with the application (e.g., because a button is not displayed correctly). Examples of minor problems are a high latency when loading data from the server or dialogues that are hard to understand.

A crucial assessment aspect is the scope, difficulty, and complexity of the functionality covered by your project/application. If you are developing a very simplistic and small app with little complexity/difficulty, it will influence your assessment. However, note that rather than aiming for a very complex and large-scope application that does not work smoothly, focus on something that works well and has a reasonable scope/complexity.

Team Collaboration Reflection Tasks
To support your group work, you are required to participate in the weekly reflection tasks after your TA meeting, which will be sent out each Monday. These tasks serve to assist your collaboration and as a basis for reflection throughout and at the end of the semester.

Continous Progress
During Milestone 4, each team member has to contribute to the project continuously. More specifically, every member has to complete at least 2 meaningful tasks per week, where a single devel-

opment task should have a granularity of 0.5-1 day. The completed tasks have to be shown in the weekly TA meetings, and you are required to ensure traceability of your work by linking git commits to development tasks. You have one "Joker" to miss one weekly TA meeting and another "Joker" to once skip continuous progress over the remaining weeks of the course. Please note that you cannot make up for "missed" continuous progress, but you can "work ahead" by completing twice the amount of work in one week to skip progress on a subsequent week without using your "Joker". Please communicate your planning ahead of time.

Note: If a team member fails to show continuous progress, they will individually fail the milestone (unless there is a valid reason).

Brownie Points
In addition to the group assessment, we will use a "brownie points" system for which you have to distribute brownie points to your team members. The brownie points should reflect how you feel about the other team members' contribution to your learning, the assignment, and your team's performance. Distributing brownie points will be an opportunity to reward the members of your team who worked hard on your behalf. You can split the brownie points equally if you think everyone did the same. Every student has 40 brownie points to distribute to the 4 other team members (if you have only 3 team members, only distribute 30 brownie points). These brownie points will also allow us to notice any concerns in a team early on. The brownie points can decide whether individual group members pass or fail for borderline submissions.