



Software Praktikum (SoPra) - FS21

Milestone 4 - Assignment

1 General Information

The Assignment has to be done as a team effort where each member is able to answer content-related questions during the presentation for Milestone 4. The hand-in for Milestone 4 consists of a written report, a presentation, and the source code (including the README file). The report and the presentation must be submitted to OLAT. The source code will be taken from the GitHub project's master branch. For all three deliverables, the deadline is **30.5.2021 23:59 CET**. The presentation is on the **31.5.2021**.

1.1 Report

The report must be submitted as a single PDF file by the **project leader via OLAT**. The title page consists of the group name, group leader, and information about all group members (name, matriculation/student number, and **GitHub account**). Make sure the report is easily readable in printed form (figures, tables, etc.). Furthermore, ensure to use a consistent format (header, footer, font style, font size, page numbers, figure and table titles, etc.), page orientation (portrait), and page size (DIN A4). Prepare the final report as if you would hand it over to the customer as project documentation. **Distinct graphics must fit on one page.** Use a file name of form **FS21-Group-GROUPNUMBER-M4-Report.pdf**. The report must be written in English.

1.2 Presentation

The final presentation should focus on the main aspects and requirements of your application. From your perspective, present the most important requirements and how you planned and implemented those. Moreover, talk about your *lessons learned* with respect to technical challenges, planning, and teamwork. Do not focus on implementation problems only, but rather include problems from all phases of the software development process. Further, describe what you have learned from SoPra and what you would do differently or better next time.

In the second part of the final presentation you are expected to give a live demo of the final version of your application. The goal of the demo is to show that your application actually complies with the requirements and runs without any bugs/problems.

Both parts have to be well prepared. A smooth presentation is what you should aim for. Therefore, a great structure, neat slides and a good preparation of the speakers is essential. A discussion follows the presentation where each group member is expected to be capable of answering questions. The question can cover topics such as implementation details, testing, project planning, etc.

The presentation must be submitted as a single PDF file by the **project leader via OLAT**. The title page consists of the group name and information about all group members (name, matriculation/student number). Use a file name of form **FS21-Group-GROUPNUMBER-M4-Slides.pdf**. The slides as well as the presentation have to be in English.

Advice: Each team member should present at least at one occasion (M1+M2, M3, or M4). It is up to the team to decide how to distribute the presentation time over its members. Presentations will be **6 minutes per team**, including the demo. After the presentation, there will be a 2-4 minutes Q&A session. Make sure to prepare your presentation and prepare slides that are readable by the audience and easy to understand. Also, ensure to practice your presentation at least once beforehand, especially to make sure you stay within the given time frame. It might be advisable to have some form of backup of your demo (e.g., screenshots or screencast) if something goes wrong when performing a live demo. Some instructors who have not seen the report will sit in the presentation, which will be your moment to shine.

1.3 Source Code

The source code submission is done through GitHub. Add a git tag **M4** to the commit in the **master** branch that should be taken into consideration for grading. It is important to push the commit before the deadline **30.5.2021 23:59 CET**. Moreover, make sure the code tagged with M4 contains a stable/running version of your system. If we can not run your system, we can not grade it and therefore must fail you. Submissions after the deadline are not taken into account.

1.4 Beta Testing

If you completed the last assignment for M3, you should have a running and stable version of your application. Now, it is time to put your application to the test. Each group will beta-test the implementation of another group. To do so, schedule a 1-hour time slot with your team members in the first week of Milestone 4. You will find the project your group is supposed to provide feedback for in the OLAT forum. **Please take approximately 30 minutes to play the game or experience the application. In the remaining time, compile short, written feedback to the developers of the app/game. It can be bullet points and should be around 0.5-1 A4 pages.** Do not forget to include positive parts as well as suggestions for improvement. In case you find bugs make sure you provide enough details to reproduce them whenever possible.

Please submit your feedback to sopra@lists.ifi.uzh.ch **before Sunday, 9.5.2021 23:59 CET**. We will then share your feedback with the other group.

2 Project Closing

The remaining weeks are about giving the project its final touch and preparing the final report.

2.1 Implementation of Requirements

Ensure that your application implements at least all essential features you have committed to in your project planning. Minor deviations are tolerated as long as the application fulfills its primary purpose (e.g., a game should be playable as defined by the game instructions).

The assessment differentiates between minor and major problems. A major problem would be the crash of the application. A minor problem could be a high latency when loading something from the server. **Not more than one major and four minor problems should occur during the grading.**

2.2 Usability

The usability assessment focuses on rendering errors, user guidance (e.g., do new users understand the application), and overall user experience.

The grading differentiates between major and minor problems similar to above. A major problem could be if a player can not perform an action because of the covering of a button. Minor problems are, for example, dialogues that are hard to understand. Not more than one major and four minor problems should occur during the grading.

2.3 Testing

Proper software projects are extensively tested. Define test cases for each user story. 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 associated with it. Further, every REST endpoint needs testing.

Note: test cases that are just there for achieving coverage but do not meaningfully test functionality (e.g., getters and setters) will not count.

2.4 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 files¹.

In both your GitHub repositories, create README.md files and write about the following three aspects using the Markdown notation²:

- **Introduction:** the project's aim
- **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?

¹<https://www.makeareadme.com>

²<https://guides.github.com/features/mastering-markdown/>

- **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 guide³)

You find a good template⁴, as well as a curated list of good READMEs⁵ online.

2.5 Reflection on Learnings & Challenges

Finally, in your report, focus on challenges (technical, planning, coordination). Reflect upon the last couple of months. What did you learn? What would you do differently next time? What helped your team work well together?

3 Grading

As SoPra is a pass/fail course, the grade for M4 will be pass or fail as well. Overall, you have to pass 3 out of 4 milestones. M1 and M4 have to be passed.

In order to pass M4, the report, the presentation, and your application need a positive assessment. The report and the presentation are graded based on their completeness and quality. The application/coding part is positive if all essential tasks of the iteration plans are completed, the code is deployed to Heroku, and the implemented user stories are functional. Further, in every weekly meeting, your TA will assess your group's performance by checking the status of your product backlog. *Every* team member has to implement at least two tasks per week. If a team member fails to do so, she/he will fail the milestone (unless there is a valid reason). You will receive feedback on your report and presentation, including an assessment (either pass, borderline pass, or fail).

In addition to the report assessment, we require you to award your teammates with "brownie points". This way, you can tell us who you think was the driver and contributed most to the project. Equally important, this is an opportunity for every student to be aware of their contribution level. Every student has 10 points per *other* team member (e.g., groups of 5, 4 other team members, hence 40 points); she/he can distribute the points on the other team members to their own liking. For borderline submissions, the brownie points can decide whether individual group members pass or fail. **Please submit your brownie points via OLAT by 30.5.2021 23:59 CET.**

³<https://choosealicense.com/>

⁴e.g., <https://gist.github.com/PurpleBooth/109311bb0361f32d87a2>

⁵e.g., <https://github.com/matiassingers/awesome-readme>