FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Supervisor's statement of a final thesis

Supervisor:	Ing. Zdeněk Rybola, Ph.D.
Student:	Jamaladdin Azizov
Thesis title:	ExpenseTracker - personal finance manager
Branch / specialization:	Web and Software Engineering
Created on:	23 August 2021

Evaluation criteria

1. Fulfillment of the assignment

▸ [1] **assignment fulfilled**
[2] assignment fulfilled with minor objections
[3] assignment fulfilled with major objections
[4] assignment not fulfilled

The assignment can be considered fulfilled. The main goals - design and implementation of the back-end part of the system with API - were achieved.

2. Main written part 62 / 100 (D)

The textual part of the thesis is below average. The text is well structured and contains only an acceptable amount of typographical and grammatical problems, mostly caused by the author being not an English speaker. However, the actual content of the text suffers many problems.
In the analytical part of the text, the current state of the XpenseTracker application created in the Software Team Project courses is described. The implemented functions are introduced, but the problems and missing functions originally planned are not stated, although they are references from several parts of the text. The specification of new requirements is not detailed enough, focusing mostly on explaining their purpose but not specifying enough details about the necessary behavior or usage.
In the design part, technologies and design of the solution are discussed. The choice of TypeScript and Node.JS technologies is not explained appropriately, especially when it means impementing quite a big part of the core functionality again in a different technology. Also the choice of SaaS and DaaS has unknown impact on the application design and affect the deployment rather than the design. Also, the system architecture does not reflect the planned mobile client besides the web client. The software architecture is properly designed as 3-layered, although it is described as monolithic. The diagram of the database structure is incorrectly interpreted as an UML diagram, but it is not.

The class and sequence model, meant to explain the structure of the solution on a specific example, is very poor. The class model uses incorrect notation (missing multiplicities for references between the Service and Repositories, wrong "specialization" relations. The presentation layer is completely missing. No methods visible for repositories (probably only generic methods provided by the framework are used), but no information about this principle provided in the text. Also, no inforation about dependency injection and other used principles provided. The sequence model is alsmost completely wrong: wrong order of messages; wrong parameters not matching the code; wrong notation for return calls.
The implementation chapter describes certain details of the realization. Authentication is described very briefly and I was not able to understand how it practically works. Also, the balance calculation related by SQL triggers is not explaing well enough. Why is t better than calculating the balance live? Why is it not possible to update it with each data operation? Similarly, there are other parts realized by precalculating data in the database beforehand (budgets, planned transactions), whose implementation is not discussed.
Unit testing is also described very briefly without any details about how big part of the codebase is covered, how many tests there are and what scenarios they cover (which proved very limited when looking in the code).

3. Non-written part, attachments 70/100 (C)

The server-side application XpenseTracker is an important part of the thesis. As the front-end part of the system is developed by a different student, only the server-side functionality, data persistence and API was implemented in this thesis. The resulting application provides most of the expected functions and can be used by the mobile and web client to handle the data and functions.
The architecture of the solution does not correspond to the model described in the text of the thesis. The code is divided into modules for invidiual entities, containing both the controller, service, entity classes and DTO classes. The code is not documented and well structured. There are many extremely complicated methods that should be better divided into several smaller methods orchastrated together. The persistence layer is whole implemented using the generic methods of the framework without any explicit declaration of domain-specific methods.
Parts of the code are covered by unit tests, but only the AccountService class contains serious tests.
API is a substantial part of the system. It is documented using Swagger, but it contains no descriptions and explanations. Also, no domain-specific examples are provided to guide the developer to correct usage of the methods and operations.

4. Evaluation of results, publication outputs and awards 70/100 (C)

The quality of the results is mainly determined by the differences between the design and its implementation. Also, as the front-end part of the system is not yes finished and the author's back-end part could mostly be tested only via API calls, it is very difficult to state that it is correct. However, the application is in the state to be used in the FE development and the author promised to be of assistance any BE-related problems would occour.

5. Activity of the student

[1] excellent activity

[2] very good activity
▸ **[3] average activity**
[4] weaker, but still sufficient activity
[5] insufficient activity

The student's activity was average. He attended most of the regular meetings, showing progress in the development. The actual text of the thesis was created at the end with limited time, which affected its quality to some extent.

6. Self-reliance of the student

[1] excellent self-reliance
[2] very good self-reliance
▸ **[3] average self-reliance**
[4] weaker, but still sufficient self-reliance
[5] insufficient self-reliance

The student's self-reliance was good. The student implemented the application on his own and consulted the progress. However, in some parts, deeper discussion about the planned and implemented solution would lead to a better design and potentially also higher code quality.

The overall evaluation 65 /100 (D)

I consider the results of the thesis below average. The text of the thesis is well structured, but it misses some important details regarding the requirements, design model and realization details. Also, the implementation does not reach the expected quality, especially regarding the code structure, documentation and unit testing.

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

Activity of the student

From your experience with the course of the work on the thesis and its outcome, review the student's activity while working on the thesis, his/her punctuality when meeting the deadlines and whether he/she consulted you as he/she went along and also, whether he/she was well prepared for these consultations.

Self-reliance of the student

From your experience with the course of the work on the thesis and its outcome, assess the student's ability to develop independent creative work.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.