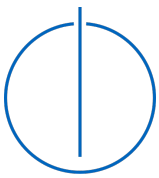
FAKULTÄT FÜR INFORMATIK

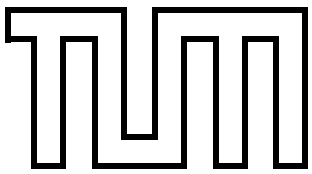
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatik

Further development of GBS Tutoring Tool: Implementation of Student facing features and functions-verified tests

Wenjie Hou





FAKULTÄT FÜR INFORMATIK

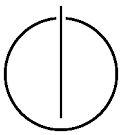
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatik

Further development of GBS Tutoring Tool: Implementation of Student facing features and functions-verified tests

Weiterentwicklung des GBS Tutoring Tools: Implementierung von studentenorientierten Features und Funktionenverifizierte Tests

Author:	Wenjie Hou
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	Dipl.-Inf. (Univ.) Martin Uhl
Date:	15.05.2021



I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 15.05.2021 Wenjie Hou

Acknowledgements

Abstract

Note:

1. paragraph: *What is the motivation of your thesis? Why is it interesting from a scientific point of view? Which main problem do you like to solve?*

2. paragraph: *What is the purpose of the document? What is the main content, the main contribution?*

3. paragraph: *What is your methodology? How do you proceed?*

Zusammenfassung

Note: Insert the German translation of the English abstract here.

Contents

GUI Graphical User Interface

Chapter 1

Introduction

Note: Introduce the topic of your thesis, e.g. with a little historical overview.

1.1 Problem

Note: Describe the problem that you like to address in your thesis to show the importance of your work. Focus on the negative symptoms of the currently available solution.

1.2 Motivation

Note: Motivate scientifically why solving this problem is necessary. What kind of benefits do we have by solving the problem?

1.3 Objectives

Note: Describe the research goals and/or research questions and how you address them by summarizing what you want to achieve in your thesis, e.g. developing a system and then evaluating it.

1.4 Outline

Note: Describe the outline of your thesis

Chapter 2

Background

Note: Describe each proven technology / concept shortly that is important to understand your thesis. Point out why it is interesting for your thesis. Make sure to incorporate references to important literature here.

2.1 e.g. User Feedback

Note: This section would summarize the concept User Feedback using definitions, historical overviews and pointing out the most important aspects of User Feedback.

2.2 e.g. Representational State Transfer

Note: This section would summarize the architectural style Representational State Transfer (REST) using definitions, historical overviews and pointing out the most important aspects of the architecture.

2.3 e.g. Scrum

Note: This section would summarize the agile method Scrum using definitions, historical overviews and pointing out the most important aspects of Scrum.

Chapter 3

Related Work

*Note: Describe related work regarding your topic and emphasize your (scientific) contribution in **contrast** to existing approaches / concepts / workflows. Related work is usually current research by others and you defend yourself against the statement: "Why is your thesis relevant? The problem was already solved by XYZ." If you have multiple related works, use subsections to separate them.*

Chapter 4

Requirements Analysis

Note: This chapter follows the Requirements Analysis Document Template in [BD09]. ***Important:*** *Make sure that the whole chapter is independent of the chosen technology and development platform. The idea is that you illustrate concepts, taxonomies and relationships of the application domain independent of the solution domain! Cite [BD09] several times in this chapter.*

4.1 Overview

Note: Provide a short overview about the purpose, scope, objectives and success criteria of the system that you like to develop.

4.2 Current System

Note: This section is only required if the proposed system (i.e. the system that you develop in the thesis) should replace an existing system.

4.3 Proposed System

Note: If you leave out the section "Current system", you can rename this section into "Requirements".

4.3.1 Functional Requirements

Note: List and describe all functional requirements of your system. Also mention requirements that you were not able to realize. The short title should be in the form "verb objective"

FR1 **Short Title**: Short Description.

FR2 **Short Title**: Short Description.

FR3 **Short Title**: Short Description.

4.3.2 Nonfunctional Requirements

Note: List and describe all nonfunctional requirements of your system. Also mention requirements that you were not able to realize. Categorize them using the FURPS+ model described in [BD09] without the category ***functionality*** *that was already covered with the functional requirements.*

NFR1 **Category**: Short Description.

NFR2 **Category**: Short Description.

NFR3 **Category**: Short Description.

4.4 System Models

Note: This section includes important system models for the requirements analysis.

4.4.1 Scenarios

Note: If you do not distinguish between visionary and demo scenarios, you can remove the two subsubsections below and list all scenarios here.

Visionary Scenarios

Note: Describe 1-2 visionary scenario here, i.e. a scenario that would perfectly solve your problem, even if it might not be realizable. Use our scenario description template (./tex/use-case-table.tex) in form of free text description.

Demo Scenarios

Note: Describe 1-2 demo scenario here, i.e. a scenario that you can implement and demonstrate until the end of your thesis. Use our scenario description template (./tex/use-case-table.tex) in form of free text description.

4.4.2 Use Case Model

Note: This subsection should contain a UML Use Case Diagram including roles and their use cases. You can use colors to indicate priorities. Think about splitting the diagram into multiple ones if you have more than 10 use cases. ***Important:*** *Make sure to describe the most important use cases using the use case table template (./tex/use-case-table.tex). Also describe the rationale of the use case model, i.e. why you modeled it like you show it in the diagram.*

4.4.3 Analysis Object Model

Note: This subsection should contain a UML Class Diagram showing the most important objects, attributes, methods and relations of your application domain including taxonomies using specification inheritance (see [BD09]). Do not insert objects, attributes or methods of the solution domain. ***Important:*** *Make sure to describe the analysis object model thoroughly in the text so that readers are able to understand the diagram. Also write about the rationale how and why you modeled the concepts like this.*

4.4.4 Dynamic Model

*Note: This subsection should contain dynamic UML diagrams. These can be a UML state diagrams, UML communication diagrams or UML activity diagrams.****Important:*** *Make sure to describe the diagram and its rationale in the text.* ***Do not use UML sequence diagrams.***

4.4.5 User Interface

Note: Show mockups of the user interface of the software you develop and their connections / transitions. You can also create a storyboard. ***Important:*** *Describe the mockups and their rationale in the text.*

Chapter 5

System Design

Note: This chapter follows the System Design Document Template in [BD09]. You describe in this chapter how you map the concepts of the application domain to the solution domain. Some sections are optional, if they do not apply to your problem. Cite [BD09] several times in this chapter.

5.1 Overview

Note: Provide a brief overview of the software architecture and references to other chapters (e.g. requirements analysis), references to existing systems, constraints impacting the software architecture.

5.2 Design Goals

Note: Derive design goals from your nonfunctional requirements, prioritize them (as they might conflict with each other) and describe the rationale of your prioritization. Any trade-offs between design goals (e.g., build vs. buy, memory space vs. response time), and the rationale behind the specific solution should be described in this section

5.3 Subsystem Decomposition

Note: Describe the architecture of your system by decomposing it into subsystems and the services provided by each subsystem. Use UML class diagrams including packages / components for each subsystem.

5.4 Hardware Software Mapping

Note: This section describes how the subsystems are mapped onto existing hardware and software components. The description is accompanied by a UML deployment diagram. The existing components are often off-the-shelf components. If the components are distributed on different nodes, the network infrastructure and the protocols are also described.

5.5 Persistent Data Management

Note: Optional section that describes how data is saved over the lifetime of the system and which data. Usually this is either done by saving data in structured files or in databases. If this is applicable for the thesis, describe the approach for persisting data here and show a UML class diagram how the entity objects are mapped to persistent storage. It contains a rationale of the selected storage scheme, file system or database, a description of the selected database and database administration issues.

5.6 Access Control

Note: Optional section describing the access control and security issues based on the nonfunctional requirements in the requirements analysis. It also describes the implementation of the access matrix based on capabilities or access control lists, the selection of authentication mechanisms and the use of encryption algorithms.

5.7 Global Software Control

Note: Optional section describing the control flow of the system, in particular, whether a monolithic, event-driven control flow or concurrent processes have been selected, how requests are initiated and specific synchronization issues

5.8 Boundary Conditions

Note: Optional section describing the use cases how to start up the separate components of the system, how to shut them down, and what to do if a component or the system fails.

Chapter 6

Case Study / Evaluation

Note: If you did an evaluation / case study, describe it here.

6.1 Design

Note: Describe the design / methodology of the evaluation and why you did it like that. E.g. what kind of evaluation have you done (e.g. questionnaire, personal interviews, simulation, quantitative analysis of metrics, what kind of participants, what kind of questions, what was the procedure?

6.2 Objectives

Note: Derive concrete objectives / hypotheses for this evaluation from the general ones in the introduction.

6.3 Results

Note: Summarize the most interesting results of your evaluation (without interpretation). Additional results can be put into the appendix.

6.4 Findings

Note: Interpret the results and conclude interesting findings

6.5 Discussion

Note: Discuss the findings in more detail and also review possible disadvantages that you found

6.6 Limitations

Note: Describe limitations and threats to validity of your evaluation, e.g. reliability, generalizability, selection bias, researcher bias

Chapter 7

Summary

Note: This chapter includes the status of your thesis, a conclusion and an outlook about future work.

7.1 Status

Note: Describe honestly the achieved goals (e.g. the well implemented and tested use cases) and the open goals here. if you only have achieved goals, you did something wrong in your analysis.

○

◐

●

7.1.1 Realized Goals

Note: Summarize the achieved goals by repeating the realized requirements or use cases stating how you realized them.

7.1.2 Open Goals

Note: Summarize the open goals by repeating the open requirements or use cases and explaining why you were not able to achieve them. ***Important:*** *It might be suspicious, if you do not have open goals. This usually indicates that you did not thoroughly analyze your problems.*

7.2 Conclusion

Note: Recap shortly which problem you solved in your thesis and discuss your ***contributions*** *here.*

7.3 Future Work

Note: Tell us the next steps (that you would do if you have more time. be creative, visionary and open-minded here.

Appendix A

e.g. Questionnaire

Note: If you have large models, additional evaluation data like questionnaires or non summarized results, put them into the appendix.

List of Figures

List of Tables

Bibliography

[BD09] Bernd Bruegge and Allen H Dutoit. *Object Oriented Software Engineering Using UML, Patterns, and Java*. Prentice Hall, 2009.