VILNIAUS KOLEGIJA HIGHER EDUCATION INSTITUTION
FACULTY OF ELECTRONICS AND INFORMATICS
SOFTWARE DEVELOPMENT DEPARTMENT

APPROVED
by the Dean of the Faculty of Electronics and Informatics of Vilniaus Kolegija Higher Education Institution, Order No. EI V2-35
November 6, 2024

METHODOLOGICAL GUIDELINES FOR THE FINAL PROJECT OF THE SOFTWARE ENGINEERING STUDY PROGRAMME

PREPARED BY
The academic staff of the Software
Development Department

2024

TABLE OF CONTENTS

INTRODUCTION 3

1. STRUCTURE OF THE FINAL PROJECT 3

1.1. Software (Project) Part of the Final Project 4

1.2. Theoretical Part of the Final Project 4

1.2.1. Introduction 5

1.2.2. Task formulation 6

1.2.3. Task analysis 6

1.2.4. Software Implementation 6

1.2.5. User manual 7

1.2.6. Conclusions and Recommendations 7

1.2.7. References 7

1.2.8. Annexes 8

2. FINAL PROJECT PREPARATION PLAN AND MONITORING 9

3. FINAL PROJECT DEFENCE AT FINAL PROJECT DEFENCE COMMITTEE 11

REFERENCES 12

ANNEXES 13

ANNEX 1. FINAL PROJECT TITLE PAGE IN ENGLISH LANGUAGE 14

ANNEX 2. ASSIGNMENT DECLARATION 15

ANNEX 3. SUMMARY IN ENGLISH LANGUAGE 17

ANNEX 4. COMPANY FEEDBACK FORM 18

ANNEX 5. LEARNING OUTCOMES OF THE SOFTWARE ENGINEERING STUDY PROGRAMME 19

INTRODUCTION

The Methodological Guidelines are issued for the students of the Faculty of Electronics and Informatics of Vilniaus Kolegija Higher Education Institution, Software Engineering study programme (state code 6531BX028). The document establishes the principles for the Final Project preparation. The step-by-step plan and defence procedure for the Final Project preparation is in accordance with the Final Project Preparation and Defence Procedure Description at Vilniaus Kolegija Higher Education Institution (2022).

1. STRUCTURE OF THE FINAL PROJECT

Students from the Software Development Department of the Faculty of Electronics and Informatics of Vilniaus Kolegija Higher Education Institution accomplish their studies by preparing and defending the Final Project.

Undergraduates establish during the study years acquired theoretical and practical knowledge by implementing applied projects. Undergraduates develop a software product according to the software and hardware defined in the task of the Final Project and prepare the theoretical part of the Final Project regarding the assigned task.

The Final Project is a qualification work where students demonstrate a level of professional preparation and reveal the learning outcomes of the Final Project.

The Final Project can be prepared by a group of 2-3 undergraduates. In this case, each undergraduate prepares a separate theoretical part corresponding to the task.

The Final Project consists of:

- Software part (Project part);
- Theoretical part.

The Final Project (theoretical and project parts) must be stored in the e-file repository of Vilniaus Kolegija Higher Education Institution. The student must submit the following documents to the e-file repository:

- **The theoretical part** of the Final Project in PDF format, file name: *N. Surname FP (Final Project).pdf;*
- **Source code**, file directory called *Source code*, with a user manual on how to deploy and release it, called *Read Me*.
- If necessary, the students can prepare video material to explain the specifics of program performance further, provide extra relevant information, etc. File name: *Software presentation video*.

A graduate who has prepared a Final Project on behalf of a company, institution, or organisation must submit written feedback from that the company, institution, or organization to the Department before the defence of the Final Project (Annex 4).

1.1. Software (Project) Part of the Final Project

The software (project) part can be:

- A program which addresses applied tasks intended for one or a group of users.
- Mobile application.
- Website.
- Software for embedded devices and data stream control.
- A distinct or group of developed software sections is intended for a larger and global project.

The software (project) part of the Final Project should not be:

- Parts of the already successfully defended Final Project.
- Designed from existing software without using any code snippets or fragments of the undergraduate source code.
- Created with software not used at the Vilniaus kolegija Higher Education Institution, the student can only freely dispose of the license of the chosen software if it is possible to demonstrate the created software on a virtual or remote computer.

Minimum requirements for the software:

- The program must function correctly;
- The graduate must develop the software code. Application code from standard libraries, frameworks, and generators (or other sources) must be revised.

1.2. Theoretical Part of the Final Project

The theoretical part of the Final Project is prepared following *the General Requirements for Written Academic Papers*, approved by the Dean of the Faculty of Electronics and Informatics of Vilniaus Kolegija Higher Education Institution. Essential information on the *General Requirements for Academic Papers* is publicly available on the Faculty of Electronics and Informatics methodological guidelines website page.

The theoretical part of the Final Project is comprised of the following structural elements:

- Title page in the English language;
- Assignment declaration of the Final Project;
- Summary in the English language;
- List of terminology definitions used and abbreviations;
- Table of contents;
- List of illustrations;

- List of tables;
- Introduction;
- Task formulation;
- Task analysis;
- Software realisation description;
- User manual;
- Conclusions and recommendations;
- References;
- Annexes (annex called ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME is mandatory).

Students must submit the printed version of the Final Project assignment declaration sheet on both sides of the sheet. The theoretical part of the Final Project needs to be printed. The final version of the theoretical part of the Final Project is stored in the DATA REPOSITORY of Vilniaus Kolegija Higher Education Institution.

Minimum requirements for the theoretical part of the Final Project:

- The theoretical part must be prepared following the General Requirements for Study Papers of the Faculty of Electronics and Informatics (Gžegoževskė et al., 2023);
- Must comply with the valid spelling and punctuation standards of the Lithuanian language;
- The scope of the theoretical part is not less than that specified in each section of the theoretical part.

A committee of three faculty members from the Department of Software Development, appointed by order of the Dean, decides on compliance with the minimum requirements of the theoretical part of the final project for the defence in the department.

1.2.1. Introduction

The Introduction consists of:

- Relevance of the topic. Undergraduates should justify why the subject has been chosen for analysis, as well as the level of its novelty and relevance, and indicate the reasons for choosing that topic. The chosen topic must comply with the specific study field. Students are recommended to provide analytical analysis of similar systems.
- Aim and objectives. In this step of the Final Project, it is necessary to disclose the objective of this final project. The aim should be realistic, measurable, and achievable in time for project preparation. The purpose should comply with the desired result. The objectives should detail the aim and cover the principal analyzed aspects.

- Implementation tools of the Final Project. In this step of the Final Project, the undergraduates provide a list of chosen software and hardware.

If a group of undergraduates prepares the final project, then the introduction should include a work plan presenting task allocation, task completion deadlines, the contribution made by each undergraduate to the task, and the use of shared resources.

The length of the introduction part is two to five pages.

1.2.2. Task formulation

The task formulation section outlines both the functional and non-functional requirements for the intended software implementation. Functional requirements detail what the software will be able to accomplish, specifying the primary and supporting functions of the application to be developed. The primary functions are designed to fulfil the main objectives of the application, while supporting functions are influenced by technological requirements. Typically, these auxiliary functions are employed for tasks such as job logging, data archiving, and statistics collection, which serve to maintain the software implementation.

The formulation of functional requirements for each function includes the initial data, the actions the function will perform, and the expected results. It should also specify the sequence in which the functions should be executed and any constraints that may apply to their implementation.

Non-functional requirements, on the other hand, are those that impose limitations on the range of possible design solutions.

This section should be 3 to 5 pages long.

1.2.3. Task analysis

Exploratory data analysis should be performed in the task analysis part:

- Use a case diagram and its description.
- Activity diagram and its report.
- Other UML diagrams and their descriptions.
- ER diagram (or its alternative) and its description.
- Class diagram (project directory structure or its option) and its report.

The length of the task analysis section is approximately ten pages.

1.2.4. Software Implementation

The student should describe software implementation files in this section, revealing their purpose. The description of the classes and methods, actions, initial data, and results structure is given. The physical model of the database (if any) and its description are presented. Other software constructions, such as components, modules, and relationships, should be described in detail.

Software code fragments must be uploaded using C*ourier New* typeface. Uploading screenshots of the code is prohibited.

The length of the software implementation section is approximately fifteen pages.

1.2.5. User manual

In this section, the undergraduate should provide an implementation guide:

- software implementation dependence on other software products (to describe the system or other processes without which system software implementation components are not able to perform);
- identify computer hardware parameters on which software implementation was completed and tested;
- a detailed description of software implementation;
- a description of typical configuration (if any);
- a detailed description of software implementation (especially implementing mobile apps, websites or other services based on Internet technologies);
- steps to eliminate software implementation.

In addition, the undergraduate should provide a user manual of software implementation - a description of the steps that address basic functional requirements.

Provide a reasoned explanation if a specific part of the user manual is missing.

The length of the section is approximately ten pages.

1.2.6. Conclusions and Recommendations

In this section, the undergraduate should formulate the conclusions directly related to the Final Project. The findings should be substantiated, specific, and related to the aim and objectives addressed in the Final Project. If the undergraduate failed to achieve the expected results or to solve all addressed purposes, it is necessary to state the reasons.

In this part, the undergraduate should also present development possibilities and ways for improving the program.

In case a group of undergraduates performed the Final Project, the contribution made by each undergraduate to the project and performed tasks should be presented.

The length of the section is approximately one to two pages.

1.2.7. References

A list of references shall only include the sources and literature used in the Final Project by the author. Information sources may include:

- books;
- periodicals;

- electronic information sources, etc.

At least seven information sources should be listed in the list of references.

1.2.8. Annexes

Annexes may be any information not directly related to the Final Project or refer to information which requires a change in the formatting requirements of the text for its representation. Mandatory annexes:

- ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME;

Annex ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME is mandatory (Annex 5). Undergraduates must achieve not less than 50% of learning outcomes (following the *Description of Assessment Procedure of Academic Achievements* approved by the Rector of Vilniaus Kolegija Higher Education Institution, 2024) The minimum number of the acquired learning outcomes of the study programme that the undergraduate must demonstrate during the Final Project preparation is at least nine such as:

- from A section "Knowledge and its application"– not less than three learning outcomes;
- from B section "Research skills" – at least one learning outcome;
- from C section "Special skills" – not less than three learning outcomes;
- from D section "Social skills" – at least one learning outcome;
- from E section "Personal skills" – at least one learning outcome.

Annexes may be:

- written consent of the institution to use its name in the theoretical part of the Final Project (if necessary) for software implementation (a free form of the template);
- letter from the institution on the planned or actual use of the results of the final project (a free form of the template);
- examples of software implementation results;
- examples of software implementation testing results;
- figures, tables, pictures, or other relevant information that was mentioned in the final project description but not presented;
- copies of articles published by the undergraduate;
- conference certificates;
- other.

2. FINAL PROJECT PREPARATION PLAN AND MONITORING

The work plan, the writing process, and other paper-development-related issues shall be discussed with the academic advisor, technical advisor and English language consultants.

The Final Project preparation monitoring includes departmental approvals and defence at the department meeting. The undergraduate must get approval on certain sections of the Final Project from the academic advisor before departmental previews.

The undergraduate who has yet to defend the project at the Software Development Department meeting loses the right to defend the Final Project at the Final Project Defence Committee meeting. Information regarding the Final Project preparation and monitoring is available and updated on the VMA Moodle (e-course title *Baigiamieji darbai_PROGRAMŲ SISTEMOS / Final Project_SOFTWARE ENGINEERING 20../20.. m.m*).

In cases of academic integrity violations, students will face sanctions in accordance with the laws of the Republic of Lithuania and the regulations governing study procedures at Vilnius Higher Education Institution.

The Final Project Preparation stages:

1. Introductory preview

The step-by-step plan is presented during this stage of the Final Project preparation.

2. First preview

At this stage, the undergraduate submits the Theoretical part of the Final Project: title page, introduction, task formulation and task analysis. Remarks and comments on the submitted parts should be considered for the following preview. The first review is conducted with the Final Project supervisor.

3. Second preview

At this stage, the undergraduate submits edited parts of the first preview and presents the basic functionality of software implementation. Remarks and comments on the submitted parts should be considered for the following preview.

4. Third preview

At this stage, the undergraduate submits corrected parts of the second preview. The undergraduate presents the following parts of the theoretical part: software implementation, implementation guidelines and user manual, conclusions and recommendations, the list of references, annexes, and summary. The undergraduate presents a complete software realisation plan. All remarks and comments on the final project should be considered till defence in the department meeting.

5. Defence at the department meeting

The undergraduate should submit the edited third preview version of the Final Project for the defence at the department meeting and finish the theoretical part and software implementation part of the Final Project. The undergraduate has to upload the theoretical part of the Final Project in Word format called *N.Surname FP theoretical part*. The date for uploading the file will be scheduled during the third preview of the Final Project.

The undergraduate must present the aim, objectives, ideas, work plan and progress, results, conclusions, and recommendations. During defence at the department meeting, the undergraduate should reveal the learning outcomes achieved from the Final Project by presenting the project description and software implementation to the department members. The department members decide if the undergraduate can defend the Final Project at the Final Project Defence Committee meeting. After the defence, undergraduates are introduced with meeting minutes.

The undergraduate who has successfully defended the Final Project at the department meeting acquires permission to upload the final project's theoretical part to the ESAS system for anti-plagiarism checks. The undergraduate must upload the Final Project by the scheduled date in the step-by-step plan of the Final Project Preparation Plan. The academic advisor presents the report of the computer check for plagiarism by the ESAS plagiarism checker to the Department.

Following the Software Development department meeting, if there is no plagiarism (more information about plagiarism is available in *the General Requirements for Written Academic Papers* document, p.14), the undergraduate can defend the Final Project at the Final Project Defence Committee meeting. The Final Project files are uploaded to the DATA REPOSITORY by the responsible person/undergraduate student:

- The theoretical part of the Final Project (the responsible person submits from the VMA Moodle platform);
- The source code, a directory called Source code, with the starting instructions, called *Read Me* (uploaded by the undergraduate);
- if necessary, video material can be prepared to explain the specifics of how the program works, provide additional important information, etc. The file name of the video materia: Software presentation video (uploaded by the undergraduate).

The department hands the work for review.

The undergraduate who has yet to defend the Final Project at the department meeting has the right to defend it next year following the studies procedure regulations (2024) of Vilniaus kolegija Higher Education Institution.

3. FINAL PROJECT DEFENCE AT FINAL PROJECT DEFENCE COMMITTEE

The Final Project Defence Committee is granted access to the DATA STORAGE to get acquainted with the reviews submitted by the reviewers before the Defence Committee meeting. During the defence, the undergraduate demonstrates the slideshow and presents the developed software (project part).

The Final Project and its defence are evaluated by marks. The lowest mark is 5, and the highest mark is 10. The final mark is the average of the marks given by the Final Project Defence Committee members and the mark given by the reviewer (following *Final Project Preparation and Defence Procedure Description* at Vilniaus Kolegija Higher Education Institution (2022).

REFERENCES

1. Elektronikos ir informatikos fakultetas metodiniai nurodymai. (n.d.). *Bendrieji studijų rašto darbų reikalavimai.* Metodiniai nurodymai – Elektronikos ir informatikos fakultetas (viko.lt)
2. Gžegoževskė, L., Kirdeikienė, A., Mačėnienė, J., Neverbickaitė, D., & Zailskas, J. (2023). *Bendrieji studijų rašto darbų reikalavimai.* https://eif.viko.lt/media/uploads/sites/5/2015/03/Bendrieji-studij%C5%B3-ra%C5%A1to-darb%C5%B3-reikalavimai2023.pdf
3. *Vilniaus kolegijos baigiamųjų darbų (projektų) rengimo ir gynimo tvarkos aprašas, patvirtintas 2022 m. spalio 12 d. Vilniaus kolegijos Akademinės tarybos nutarimu Nr. AT N – 12*. (2022). Vilniaus kolegija. https://www.viko.lt/wp-content/uploads/sites/8/2022/10/BD_tvarkos_aprasas_2022-10-12_GALUTINIS.pdf
4. *Vilniaus kolegijos studijų tvarka, patvirtinta 2023 m. birželio 14 d. Vilniaus kolegijos Akademinės tarybos nutarimu Nr. AT N-6*. (2023). Vilniaus kolegija. https://www.viko.lt/wp-content/uploads/sites/8/2023/06/VIKO_Studiju_tvarka_su-pakeitimais_nuo_2023-06-14.pdf

ANNEXES

ANNEX 1. FINAL PROJECT TITLE PAGE IN ENGLISH LANGUAGE

VILNIAUS KOLEGIJA HIGHER EDUCATION INSTITUTION
FACULTY OF ELECTRONICS AND INFORMATICS

FINAL PROJECT TITLE

FINAL PROJECT
FP 6531BX028 PI20X

UNDERGRADUATE FULL NAME

__/__/20__

ACADEMIC SUPERVISOR FULL NAME

__/__/20__

20__

ANNEX 2. ASSIGNMENT DECLARATION

VILNIAUS KOLEGIJA HIGHER EDUCATION INSTITUTION
FACULTY OF ELECTRONICS AND INFORMATICS

APPROVED BY
Vice-Dean of the Faculty of Electronics
and Informatics

____

October ___, ____

FINAL PROJECT ASSIGNMENT FORM

Assigned to undergraduate **Name Surname** of group **PI**_____ on *October* __, 20__.

Title of the Final Project: Write the Final Project Title

Final Project Title: Write Actual Final Project Title
The Objective of the Final Project: A brief, clear, one-sentence description focused on the software being developed. For example: Create, update ….
Final Project Tasks: list preliminary tasks that will help achieve the objective. The tasks should clearly define the functionality of the software being developed.
Tools for Final Project Implementation: List the tools and technologies that will be used in the development of the software. For example: Visual Studio, .NET Core, SQL Server, ReactJS, GitHub.

The Final Project will be defended in the meeting of the Software Development department on *January* __, 20__.

Undergraduate…….……………………………………………………………………………

(Signature) (name, surname)

Academic supervisor...

(Signature) (name, surname)

Approved:

Software Development Department Head......………..........………………..

(Signature) (name, surname)

Technical advisors:

....……………..

(Signature) (name, surname)

.............…………...

(Signature) (name, surname)

English language advisor:

....……………...

(Signature) (name, surname)

ANNEX 3. SUMMARY IN ENGLISH LANGUAGE

SUMMARY

Vilniaus Kolegija Higher Education Institution
Faculty of Electronics and Informatics
Department of Software Development

Study Programme: Software Engineering, state code– 6531BX028

Title of the Final Project: **TITLE**
Undergraduate **NAME, SURNAME**
Academic supervisor **NAME, SURNAME**

Length of the Final Project – .... p. text without annexes, ... illustrations, ... tables, ... references, ... annexes.

The summary concisely describes the essential points covered by the Final Project and is at least one page long. In the summary, it is necessary to briefly introduce the main issues of each core part of the paper, to describe the research methodology, to discuss the research results obtained, present the solutions to the problem delineated in the project part, and to draw conclusions.

Summary length is 1 page.

Keywords: ....

ANNEX 4. COMPANY FEEDBACK FORM

COMPANY INFORMATION

Vilniaus Kolegija Higher Education Institution
Faculty of Electronics and Informatics
Software Development Department

FEEDBACK

December/January __, 20__
Vilnius

We hereby confirm that the student (*Name, Surname*) ...., enrolled in the Software Engineering study programme, addressed a relevant problem in the Final Project titled .... (*Final Project title*) in accordance with the company's request.

________________	________________	________________
Position held	Signature	First name, Surname

ANNEX 5. LEARNING OUTCOMES OF THE SOFTWARE ENGINEERING STUDY PROGRAMME

Description of the study learning outcomes for the study cycle		**Study programme learning outcomes**		**Justification**
A.	Knowledge and its application.	**A.1**	To explain the fundamental facts, concepts, theories, and mathematical methods related to the computer performance, computer hardware and software, their characteristics and possibilities of practical application, computer communication and applicable solutions associated with the important events in the past, present, and predicted trends in the future of computing science.	
		A.2	To explain the principles of algorithm design, specifications of technical and functional requirements, the structure of computer programming languages and technologies, the principles of human-computer interaction, and traditional phases of the Software Development Life Cycle and software development and maintenance methods.	
		A.3	To explain how business, industrial, economic and social context interacts with professional training activity in accordance with ethical and legal requirements such as data protection, intellectual property rights, agreements, product safety, responsibilities and other associated issues.	
		A.4	To apply the study subject knowledge of software engineering field of study for developing safe products which meet specific criteria of computing science applications to solve the problems related to the field of professional activity.	
		A.5	To explain the specification for the program systems, design, testing and documentation, program system processes, models and methods.	
B.	Research skills	**B.1**	To describe the problem related to the field of professional activity in terms of the database systems, internet technologies, smart devices programming and prepare the required data and information from various sources to solve the specific problem related to the field of professional activity.	

		B.2	To analyze and evaluate data and information necessary to solve the specific problem of the professional activity related to the database systems, internet technologies, smart devices programming and justify the solutions with argumentative conclusions.	
C.	Special skills	**C.1**	To apply the Software Development Life Cycle models, software development, maintenance and project management methods, standards, development environments and tools, programming paradigms and algorithms in standard software application projects.	
		C.2	To select the right software development and maintenance tools implemented in the Software Development Life Cycle and project management.	
		C.3	To design software architecture, components, graphical user interface and testing programs using functional and non-functional requirements applied for the system.	
		C.4	To prepare specification, project and other documentation necessary to create, deploy, develop, use and administer software product or service.	
		C.5	To realize software product or service for solving the specific problem related to field of professional activity using functional and non-functional requirements applied for software.	
		C.6	To test the quality of software, its separate components and graphical user interface.	
D.	Social skills	**D.1**	To professionally communicate in state and at least in one foreign language with professional audience.	
		D.2	To work in teams in accordance with professional, ethical behavior and social responsibility principles and rules.	
E.	Personal skills	**E.1**	To self-study and work seeking for personal and professional lifelong development.	
		E.2	To demonstrate creativity for solving the tasks and problems related to the professional activity.	