



VILNIUS UNIVERSITY OF APPLIED SCIENCES  
FACULTY OF ELECTRONICS AND INFORMATICS  
SOFTWARE DEVELOPMENT DEPARTMENT

APPROVED

by the Dean of the Faculty of Electronics and  
Informatics of Vilnius University of Applied  
Sciences Order No. EI V2-27 of October 10,  
2022

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

PREPARED BY  
Software Development Department

## TABLE OF CONTENTS

<b>INTRODUCTION .....</b>	<b>3</b>
<b>1. STRUCTURE OF THE FINAL PROJECT.....</b>	<b>3</b>
<b>1.1. Software Design and Implementation Part of the Final Project.....</b>	<b>4</b>
<b>1.2. Theoretical Part of the Final Project.....</b>	<b>4</b>
<b>2. FINAL PROJECT PREPARATION PROGRESS AND MONITORING .....</b>	<b>8</b>
<b>3. FINAL PROJECT DEFENCE AT FINAL PROJECT DEFENCE COMMITTEE.....</b>	<b>10</b>
<b>ANNEXES .....</b>	<b>12</b>
<b>ANNEX 1. FINAL PROJECT TITLE PAGE .....</b>	<b>13</b>
<b>ANNEX 2. ASSIGNMENT DECLARATION.....</b>	<b>14</b>
<b>ANNEX 3. SUMMARY .....</b>	<b>16</b>
<b>ANNEX 4. LEARNING OUTCOMES OF THE SOFTWARE ENGINEERING STUDY     PROGRAMME .....</b>	<b>17</b>

## INTRODUCTION

The Methodological Guidelines are issued for the students of the Faculty of Electronics and Informatics of Vilnius University of Applied Sciences, Software Engineering study programme (state code 6531BX028). The document establishes the principles for the Final Project preparation. The procedure and step-by-step-plan for the Final Project preparation are in accordance with Final Project Preparation and Defence Procedure Description at Vilniaus kolegija/University of Applied Sciences (orig. Vilniaus kolegijos baigiamųjų darbų (projektų) rengimo ir gynimo tvarkos aprašas) approved by Academic Council of Vilnius University of Applied Sciences.

### 1. STRUCTURE OF THE FINAL PROJECT

Students from the Software Development department of the Faculty of Electronics and Informatics of Vilnius University of Applied Sciences accomplish their studies by preparing and defending the Final Project. In accordance with Procedure of Studies at Vilnius University of Applied Sciences (orig. Vilniaus kolegijos studijų reglamentas), only the students who completely fulfilled the study programme and do not have academic debts are eligible to prepare their Final Project.

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

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

The Final Project consists of:

- software design and implementation part (project part);
- theoretical part.

The Final Project (theoretical part and project part) is the property of Vilnius University of Applied Sciences and must be stored in e-file storage of Vilnius University of Applied Sciences. Students must upload the following files to e-file storage system no later than the set deadlines:

- **theoretical part** of the Final Project in the PDF format, file name: *N. Surname FP (Final Project) theoretical part. pdf*
- **final software/application version** available for deployment and release (all information in one file directory), file name: *Program architecture*;
- **Source code**, file directory called *Source code*;

- if it is necessary, the students can prepare a **video material** to deeper explain the specifics of program performance, provide with extra relevant information, etc. File name: *Software presentation video*.

The undergraduate, wishing to use the name of an institution (except Vilnius University of Applied Sciences) in the theoretical part and (or) software design and implementation part, for which software is being implemented, should give the consent and the letter signed by the head of the institution or authorized person regarding the planned or actual use of the results of the Final Project.

### **1.1. Software Design and Implementation Part of the Final Project**

Software design and implementation part can be:

- a program which addresses applied tasks, intended for one or a group of users;
- smart device application;
- website;
- software for embedded devices and data stream control.

The Final Project software design and implementation (project) part should not be:

- parts of already successfully defended Final Project.
- designed from already existing software without the use of any code snippets or fragments of the undergraduate source code.
- designed using software that is not used at Vilnius University of Applied Sciences and the undergraduate cannot freely dispose the license of the chosen software (except when it is possible to demonstrate designed software on a virtual or remote machine).

### **1.2. Theoretical Part of the Final Project**

Theoretical part for the Final Project are prepared in accordance with *the General requirements for written academic papers*, approved by the Rector of Vilnius University of Applied Sciences. Important information of General Requirements for Academic Papers is publicly available on the page of Vilnius University of Applied Sciences website <https://en.viko.lt/facilities-services/organisation-of-studies/>.

Theoretical part of the Final Project is comprised of the following structural elements:

- Title page;
- Assignment declaration;
- Summary;
- List of abbreviations;
- List of illustrations;

- List of tables;
- Table of contents;
- Introduction;
- Task formulation;
- Task analysis;
- Software implementation;
- Implementation guidelines and user manual;
- Conclusions and recommendations;
- List of references;
- Annexes (annex called ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME is mandatory).

Students must submit the printed version the theoretical part of the Final Project on one side of the sheet, only the cover letter must be printed on both sides of the sheet.

### **1.2.1. Introduction**

The Introduction consists of:

- Relevance of the topic. Undergraduate should justify why the topic has been chosen for analysis, as well as the level of its novelty and relevance, and to indicate the reasons of choosing that topic. It is necessary for the chosen topic to comply with the specific study field. Students are recommended to provide with analytical analysis of similar systems.
- Aim and objectives. In this step of the Final Project, it is necessary to disclose the aim of this final project. The aim should be realistic, measurable, and achievable in time given for project preparation. The aim should comply with the desired result. The objectives should detail the aim and cover the principal analysed aspects.
- Implementation tools of the Final Project. In this step of the Final Project, the undergraduates provide with a list of chosen software and hardware.

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

The length of introduction part is 2 to 5 pages.

### **1.2.2. Task Formulation**

The task formulation describes functional and non-functional requirements for software implementation. Functional requirements should define what implemented software must have to work properly. These requirements specify the main and auxiliary functions of the designed program. The main functions are intended to realize the designed program. The auxiliary functions are those

that are influenced by technological requirements. Typically, auxiliary functions are used to maintain or supervise software implementation (work reports, data archiving, statistics storage, etc.).

The undergraduate should indicate the initial data, actions performed by the function and the result when formulating functional requirements for each function. It is also necessary to specify the order of function execution and execution constraints, if any.

Non-functional requirements define requirements that limit the range of possible design choices.

This section should also provide an analytical overview of similar systems.

The length of the task formulation in the Final Project is 3 to 5 pages.

### **1.2.3. Task Analysis**

Exploratory task analysis should be performed in task analysis part:

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

The length of task analysis section is approximately 10 pages.

### **1.2.4. Software Implementation**

In this section, the student should describe software implementation files, revealing their purpose. The description of the classes and methods: actions, initial data, structure of the results is given. The physical model of the database (if any) and its description are presented. Other software constructions such as components, modules, and the relationships should be described in a detail. Software code fragments must be uploaded using *Courier New* typeface. It is prohibited to upload screenshots of the code.

The length of software implementation section is approximately 25 pages.

### **1.2.5. Implementation Guidelines and User Manual**

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

- software implementation dependence on other software products (to provide the description of 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 performed 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 the Internet technologies);
- steps to eliminate software implementation.

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

The length of the section is approximately 10 pages.

### **1.2.6. Conclusions and Recommendations**

In this section, the undergraduate should formulate the conclusions in direct relation to the Final Project. The conclusions should be substantiated, specific, 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 objectives, 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 if the Final Project was performed by a group of the undergraduates, the contribution made of each undergraduate to the project and performed tasks should be presented.

The length of the section is approximately 1 to 2 pages.

### **1.2.7. List of 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 7 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 to change formatting requirements of the text for its representation.

Annex ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME is mandatory (Annex 4). Undergraduate must achieve not less than 50% of learning outcomes (in accordance with *Description of Assessment Procedure of Academic Achievements* at Vilnius University of Applied Sciences approved by the Rector of Vilnius University of Applied Sciences, Order No. V-153, May 21, 2020.) 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 9:

- from A section “Knowledge and its application”– not less than 3 learning outcomes;

- from B section “Research skills” – at least 1 learning outcome;
- from C section “Special skills” – not less than 3 learning outcomes;
- from D section “Social skills” – at least 1 learning outcome;
- from E section “Personal skills” – at least 1 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 (free form of template);
  - letter from the institution on the planned or actual use of the results of the final project (free form of 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 PROGRESS 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 an approval on certain sections of the Final Project from academic advisor before departmental previews. The undergraduate, who has not defended the project at the Software Development department meeting, loses the right to defend the Final Project at Final Project Defence Committee meeting. Information regarding the Final Project preparation and monitoring is available and updated on the Faculty website <https://eif.viko.lt/studentams/metodiniai-nurodymai/>, and on Moodle (e-course title *Baigiamieji darbai\_PROGRAMŲ SISTEMOS / Final Project\_SOFTWARE ENGINEERING 20../20.. m.m.*).

In accordance with Studies regulations of Vilnius University of Applied Sciences, the Code of Academic Ethics of Vilnius University of Applied Sciences, the undergraduate guarantees that prepared Final Project is not plagiarized. The undergraduate who presents a plagiarized Final Project loses all rights to defend the project at Final Project Defence Committee meeting. Moreover, the undergraduate is administered according to the Republic of Lithuania Code of Administrative Offences (25 June 2015, No. XII-1869) Article 123.

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 following parts of 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 next preview.

### **3. Second preview**

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

### **4. Third preview**

At this stage the undergraduate submits corrected parts of the second preview. Submits the following parts of the Theoretical part: software implementation, implementation guidelines and user manual, conclusions and recommendations, the list of references, annexes, summary. 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, finish 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 and objectives, ideas, work plan and progress, results, conclusions and recommendations. During defence at the department meeting the undergraduate should reveal achieved learning outcomes of the Final Project by presenting project description and software implementation to the members of the department. The members of the department decide if the undergraduate is allowed to defend the Final Project at 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 the permission with access to upload the Final Project to ESAS system for anti-plagiarism checks. The undergraduate must upload the Final Project until the scheduled date in step-by-step-plan of the Final Project Preparation. The report of the computer check for plagiarism by the ESAS plagiarism checker, as well as any other computer check on student's independence in the development of the Final Project, are to be presented by the academic advisor. In case, if a plagiarism is detected, the academic advisor informs the Committee of Ethics, the Dean, head of the department and student by attaching plagiarism detection report.

In accordance with the Software Development department meeting minutes and the Dean's order, if there are no plagiarism, the undergraduate is allowed to defend the Final Project at Final Project Defence Committee meeting.

The undergraduates, who received a permission to defend the Final Project, must present the printed and signed (with necessary signatures on the title page of the undergraduate and supervisor, as well as the signatures of the responsible persons on the cover letter) Final Project. The Final Project must be not yet stapled and handed to the to the Vice-Dean. Following the minutes of the Software Development department meeting and the Dean's order, the undergraduate is awarded the right to defend the Final Project at Final Project Defence Committee and the reviewer is assigned.

The Final project must be stapled in this order:

1. Title Page;
2. Assignment Declaration;
3. Summary;
4. List of Abbreviations;
5. List of Illustrations;
6. List of Tables;
7. Table of Contents;
8. Introduction;
9. Task Formulation;
10. Task Analysis;
11. Software Implementation;
12. Implementation Guidelines and User Manual;
13. Conclusions and Recommendations;
14. List of References;
15. Annexes (annex called ACHIEVEMENT LEVEL FOR LEARNING OUTCOMES OF SOFTWARE ENGINEERING STUDY PROGRAMME is mandatory).

The undergraduate hands the stapled Final Project to the Department. The undergraduate uploads all Final Project files to E-FILE STORAGE. The Department hands the Final Project to the reviewer(s).

The undergraduate, who has not defended the Final Project at the department meeting, has the right to defend it next year in accordance with to studies procedure regulations.

### **3. FINAL PROJECT DEFENCE AT FINAL PROJECT DEFENCE COMMITTEE**

The undergraduate submits the theoretical part of the Final Project, Final Project review and presentation slideshow for the Final Project Defence Committee (hereinafter the Committee). During

the defence the undergraduate demonstrates the slideshow and presents the developed software (project part).

The Committee evaluate the Final Project by putting a grade and suggest to award the undergraduate Bachelor's degree in informatics and issue a professional Bachelor's diploma.

Final Project and its Defence are evaluated by marks. The lowest mark – 5, the highest mark – 10. The final mark is the average of the marks given by the members of the Final Project Defence Committee and the mark given by the reviewer.

## **ANNEXES**

**ANNEX 1. FINAL PROJECT TITLE PAGE**



**VILNIUS UNIVERSITY OF APPLIED SCIENCES  
FACULTY OF ELECTRONICS AND INFORMATICS**

**AUTHORIZED BY**  
Vice-Dean of the Faculty of Electronics and  
Informatics

\_\_\_\_\_PhD. Loreta Savulionienė

2023-\_\_\_\_-\_\_

**FINAL PROJECT TITLE**

**FINAL PROJECT**  
FP 6531BX028 PI19X

**UNDERGRADUATE**

**FULL NAME**

2023-\_\_-\_\_

**SUPERVISOR**

**FULL NAME**

2023-\_\_-\_\_

2023

## ANNEX 2. ASSIGNMENT DECLARATION

### VILNIUS UNIVERSITY OF APPLIED SCIENCES FACULTY OF ELECTRONICS AND INFORMATICS

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

\_\_\_\_\_ dr. Loreta Savulionienė

20\_\_ - \_\_\_\_\_ - \_\_

### FINAL PROJECT ASSIGNMENT

Given to undergraduate **Name Surname** of group **PI**\_\_\_\_\_ on October \_\_, 20\_\_ .

**Title of the Final Project:** Write Actual Final Project Title

#### Description of the Final Project

In this part of the Final Project the undergraduate should present the aim of the Final Project, relevance of the topic, the objectives related to the aim, software implementation aspects. The undergraduate must clearly present the functionality of software.

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

**Undergraduate**.....  
(signature) (name, surname)

**Supervisor**.....  
(signature) (name, surname)

Approved by:

**Head of Software Development Department**.....  
(signature) (name, surname)

**Technical advisors:**

.....  
(signature) (name, surname)

.....  
(signature) (name, surname)

**English language advisor:**

.....  
(signature) (name, surname)

**SUMMARY**

Vilnius University of Applied Sciences  
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**

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 in length. 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, to present the solutions to the problem delineated in the project part and to draw final conclusions.

Summary length is 1 page.

**Keywords:** .....



**ANNEX 4. LEARNING OUTCOMES OF THE SOFTWARE ENGINEERING STUDY PROGRAMME**

Description of the study learning outcomes for the study cycle		Study programme learning outcomes		Justification
<b>A.</b>	Knowledge and its application.	<b>A.1</b>	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.	
		<b>A.2</b>	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.	
		<b>A.3</b>	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.	
		<b>A.4</b>	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.	
		<b>A.5</b>	To explain the specification for the program systems, design, testing and documentation, program system processes, models and methods.	
<b>B.</b>	Research skills	<b>B.1</b>	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>B.2</b>	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.	
<b>C.</b>	Special skills	<b>C.1</b>	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.	
		<b>C.2</b>	To select the right software development and maintenance tools implemented in the Software Development Life Cycle and project management.	
		<b>C.3</b>	To design software architecture, components, graphical user interface and testing programs using functional and non-functional requirements applied for the system.	
		<b>C.4</b>	To prepare specification, project and other documentation necessary to create, deploy, develop, use and administer software product or service.	
		<b>C.5</b>	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.	
		<b>C.6</b>	To test the quality of software, its separate components and graphical user interface.	
		<b>D.</b>	Social skills	<b>D.1</b>
<b>D.2</b>	To work in teams in accordance with professional, ethical behaviour and social responsibility principles and rules.			
<b>E.</b>	Personal skills	<b>E.1</b>	To self-study and work seeking for personal and professional lifelong development.	
		<b>E.2</b>	To demonstrate creativity for solving the tasks and problems related to the professional activity.	