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

APPROVED
by Order of the Dean of the Faculty of Electronics and Informatics of Vilniaus kolegija
No. EI V2-30 of 9 October 2026

METHODOLOGICAL GUIDELINES FOR THE PREPARATION AND DEFENCE OF THE FINAL PROJECT IN THE SOFTWARE ENGINEERING STUDY PROGRAMME

CONSIDERED
at the meeting of the Software Development Department
Minutes No. EI K (PI)-11 of 30 September 2026

2026

TABLE OF CONTENTS

INTRODUCTION **3**

1. PURPOSE AND STRUCTURE OF THE FINAL PROJECT **3**

1.1. Software Developed for the Final Project **4**

1.1.1. Requirements for the Software **5**

1.1.2. Size of the Software Source Code.....Error! Bookmark not defined.

1.1.3. Software Solutions Unsuitable for the Final Project **6**

1.2. Final Project Report..... **7**

1.2.1. Introduction **8**

1.2.2. Task Analysis **8**

1.2.3. Software System Implementation **9**

1.2.4. Software System Testing..... **9**

1.2.5. Installation and User Manual..... **10**

1.2.6. Conclusions and Recommendations **10**

1.2.7. References..... **11**

1.2.8. Annexes..... **11**

2. ORGANISATION OF FINAL PROJECT PREPARATION, REVIEW AND DEFENCE 12

INTRODUCTION

The Methodological Guidelines for the Preparation and Defence of the Final Project in the Software Engineering Study Programme (hereinafter the Guidelines) are intended for students enrolled in the Software Engineering study programme (state code 6531BX028) at the Faculty of Electronics and Informatics of Vilniaus kolegija.

The Guidelines set out the procedures for the preparation and review of the Final Project (hereinafter FP) in the Software Engineering study programme, its defence before the Departmental Committee, its defence before the Final Project Defence Committee, and its storage. They have been prepared in accordance with the Procedure for the Preparation, Defence and Storage of Final Projects at Vilniaus kolegija.

Current information on FP preparation, review and defence is provided and regularly updated in the Moodle virtual learning environment (VLE), in the e-course *Baigiamieji darbai_PROGRAMŲ SISTEMOS / Final Project_SOFTWARE ENGINEERING*.

1. PURPOSE AND STRUCTURE OF THE FINAL PROJECT

The FP is an original piece of work prepared independently by the student to demonstrate the learning outcomes achieved in the Software Engineering study programme. In preparing the FP, students apply and consolidate the theoretical and practical knowledge, skills and competences acquired during their studies. Students must follow the information provided in the FP Topic Registration Form. The aim, objectives and other essential elements of the FP must be consistent with the information approved in that form.

The student's FP consists of two parts:

- **The software part of the FP** consists of a working software system developed by the student (hereinafter the software) to achieve the stated aim of the FP.
- **The written part of the FP** (hereinafter the report) describes the analysis, design, implementation and testing carried out. It must demonstrate that the aim of the FP has been achieved, the objectives have been met, and the software system developed meets the specified requirements.

The FP must form a coherent whole, with the software and the report complementing and supporting each other.

The FP is stored in the electronic file repository of Vilniaus kolegija. The following FP materials are stored:

- **the report** as a PDF document named *Vardas_Pavarde_Grupe*;
- **the software source code** in a folder named *Vardas_Pavarde_Source_code*, together with instructions for running the software in a file named *ReadMe*;

- **a video** explaining how the software works, the requirements met, the functions implemented and other important information. The video file must be named *Vardas_Pavarde_video*;
- **written feedback from the company, institution or organisation** if the FP was commissioned. The feedback must be prepared on the official letterhead of the company, institution or organisation.

In exceptional cases, the Dean of the Faculty (hereinafter the Dean), on the recommendation of the Head of the Software Development Department (hereinafter the Head of Department), may permit two students or a group of students to prepare and defend an FP. Each student must be assigned a clearly defined part of the FP for which they are responsible. Each student's tasks must be sufficient to demonstrate their individual achievement of the learning outcomes. Each student prepares an individual FP report. The report must clearly identify each student's individual contribution to the jointly prepared FP. During the FP defence, each student presents their contribution individually and is assessed separately on the learning outcomes they demonstrate.

1.1. Software Developed for the Final Project

The software developed for the FP must be the student's own independent work and must demonstrate the learning outcomes achieved in the Software Engineering study programme. Various types of software system may be developed, provided they are appropriate to the aim and objectives of the FP and the problem being addressed. Their suitability is assessed against the requirements in Section 1.1.1. Examples include:

- a web-based system;
- a mobile application;
- a desktop application;
- a data analysis or artificial intelligence solution;
- an Internet of Things (IoT) system;
- an embedded software system;
- a computer game;
- a programming language;
- a software framework;
- a software component or subsystem developed as part of a larger project;
- another software solution demonstrating the learning outcomes of the Software Engineering study programme.

1.1.1. Requirements for the Software

The scope, functionality and technical complexity of the software must be sufficient for the student to demonstrate the learning outcomes achieved in the Software Engineering study programme. The main requirements are set out in Table 1.

Table 1. Requirements for the Software

No.	Requirement area	Requirement description
1.	**Independent work**	The part of the software developed by the student must be clearly identifiable. The student must understand its architecture, source code and other technical solutions, and be able to explain and modify them independently. The FP report must clearly identify the student's contribution to developing the software.
2.	**Functionality**	The software must provide the functionality needed to achieve the aim of the FP and address the problem. The functions implemented must work together as a coherent solution. The functionality must be clearly specified through functional requirements and must be verifiable.
3.	**Technical complexity**	The software must demonstrate the student's ability to design and implement software systems. Technical complexity is assessed by considering the following aspects: • architecture; • algorithms and data processing; • state or process management; • integrations; • parallel or distributed execution; • hardware–software interaction; • security solutions; • other technical solutions appropriate to the nature of the software.
4.	**Architecture**	The system must have a clear architecture or modular structure.
5.	**Source code**	The software source code must be logically structured and documented. During FP preparation and defence, the source code must be stored in a version control system (e.g. *GitHub*).
6.	**Software project files**	The software project files must be submitted together with instructions for installation, configuration and running the software. Using these instructions, another person with the specified hardware and software environment must be able to set up the software for execution.
7.	**Third-party components**	Libraries, frameworks, APIs, pre-built models and other third-party components may be used, provided they do not replace the core part of the software that the student must develop independently. Their use must be justified in the FP report.
8.	**Artificial intelligence (AI) tools**	AI tools may be used to assist in the design and implementation of the software. The use of AI tools must be disclosed in the FP report in accordance with the requirements established by Vilniaus kolegija.
9.	**Testability and compliance with requirements**	It must be possible to verify the functional and non-functional requirements specified for the software. The student must be able to demonstrate compliance with those requirements through testing or other methods appropriate to the software.

No.	Requirement area	Requirement description
10.	**Software operation**	The software must function correctly and be suitable for demonstration during the FP defence.

1.1.2. Size of the Software Source Code

The scope of the software must reflect its functionality, technical complexity and the extent of the solutions implemented. In assessing its scope and technical complexity, account is taken of the type of software, its functionality and architecture, the complexity of its algorithms and data processing, interactions between components, integrations, and other relevant technical solutions.

It is recommended that the source code independently written by the student should comprise **at least 2,000 lines**. If fewer lines are submitted, this must be justified in the FP report, taking into account the type of software and the complexity of the technical solutions implemented.

The following must not be included in the line count for source code independently written by the student:

- automatically generated code;
- code from third-party libraries and other software components;
- framework templates;
- configuration and dependency specification files;
- other code or file contents that do not contain software functionality or technical solutions independently implemented by the student.

Size of the software Ssource code is considered alongside the other requirements for software suitability set out in Section 1.1.1.

1.1.3. Software Solutions Unsuitable for the Final Project

The software is considered unsuitable for the FP if it does not meet the requirements set out in Section 1.1.1 or does not enable the student to demonstrate the learning outcomes achieved in the Software Engineering study programme.

The software is considered unsuitable for the FP if:

- the student's independent contribution cannot be clearly identified;
- work previously assessed during the student's studies or as part of another FP is submitted without clearly distinguishing and justifying the newly developed part of the software;
- source code by other authors or publicly available source code constitutes the essential part of the software, and the student's contribution is limited to minor changes or adaptations (e.g. code from *GitHub*, *Stack Overflow* or other sources);
- third-party libraries, frameworks, APIs, pre-built models or other software components implement the core part of the software, and the student's contribution is limited to configuring, connecting or making minor adaptations to them;

- the essential part of the software is implemented using *low-code* or *no-code* tools, and the student's contribution is limited to configuring and connecting the components provided by those tools;
- the software is limited to a user interface implementation and contains no software logic or other significant technical solutions independently implemented by the student;
- the software is limited to basic create, read, update and delete (CRUD) operations and contains no other significant software solutions independently implemented by the student;
- the student cannot explain and justify the software architecture, source code or other essential technical solutions implemented, or cannot modify them independently;
- the operation of the software cannot be demonstrated during the FP defence.

The Departmental Committee assesses whether the software meets these Guidelines during the departmental FP defence.

1.2. Final Project Report

The FP report describes the software developed, the development process and the implementation decisions made. It must demonstrate the student's ability to design and implement a software system, justify technical decisions and demonstrate the learning outcomes achieved during their studies. The report must be prepared in accordance with the approved *General Requirements for Written Academic Papers*. The current version is available in the Moodle VLE.

The FP report consists of the following parts:

- title page in Lithuanian (template provided in Annex 1);
- title page in English (template provided in Annex 2);
- summary in Lithuanian (template provided in Annex 3; length: 1 page);
- summary in English (template provided in Annex 4; length: 1 page);
- glossary of terms and abbreviations;
- table of contents;
- list of figures;
- list of tables;
- introduction (3–5 pages);
- task analysis (at least 15 pages);
- Software system implementation (at least 15 pages);
- Software system testing (3–5 pages);
- installation and user manual (3–5 pages);
- conclusions and recommendations (1–2 pages);
- references (at least 7 sources);

- annexes (optional).

During the departmental FP defence, the Departmental Committee assesses whether the report meets these Guidelines and the General Requirements for Written Academic Papers.

1.2.1. Introduction

The introduction must reflect the information provided in the FP Topic Registration Form. It consists of:

- **Problem statement.** The student must justify the choice of FP topic and explain the relevance of the problem being addressed. The problem must be defined within its application domain, together with its practical significance and the expected benefits of the proposed solution.
- **Analysis of similar systems.** Identify the strengths and weaknesses of existing software solutions and the features relevant to the software being developed. Present the findings that inform the software requirements and design decisions.
- **Aim.** The aim must be realistic, measurable and achievable within the time allocated to preparing the FP.
- **Objectives.** Specific objectives that contribute to achieving the stated aim must be formulated. The objectives must cover the main stages of the work and the tasks to be undertaken.
- **Implementation tools.** The student provides a list of the selected software and hardware tools and justifies their choice.

If the FP is prepared by a group of students, the introduction must include a work plan showing the allocation of FP tasks among students, their responsibilities, the use of shared resources, etc.

The introduction must be 3–5 pages long.

1.2.2. Task Analysis

This section describes the software design decisions. It must include an analysis of the system requirements and models, with accompanying descriptions, of the operational processes, data structures, user interactions and architecture:

- functional requirements;
- non-functional requirements;
- a use case diagram;
- an activity diagram;
- an entity–relationship diagram (or an alternative);
- a class diagram, a project directory structure or another alternative appropriate to the architecture;
- user interface mock-ups;
- other UML diagrams.

Functional requirements specify what the software must do to address the problem. Each functional requirement must be verifiable and implemented in the system. The functionality of the software must not be confused with the objectives of the FP.

Non-functional requirements define the software's quality attributes, technological constraints and operational characteristics. Non-functional requirements must be specific, measurable and verifiable.

This section must be at least 15 pages long.

1.2.3. Software System Implementation

This section must describe the key technical solutions used to implement the software system, explain how they were implemented and show how they interact. It is recommended to include:

- descriptions of modules and components;
- the implementation of classes, components or services;
- the database implementation;
- descriptions of API methods;
- algorithm implementations;
- the implementation of integrations;
- examples of key implementation techniques, illustrated with source code excerpts.

Source code excerpts included in the FP report must be formatted in *Courier New*, 10 pt, with single line spacing. Each excerpt must not exceed half a page. Longer excerpts must be placed in the annexes. Screenshots of source code must not be included in the FP report.

This section must be at least 15 pages long.

1.2.4. Software System Testing

This section must describe the testing carried out on the software system. The testing methods are selected according to the type of software and the specified requirements. An exhaustive list of all tests or test scenarios is not required. The section should focus on the rationale for the testing methodology, significant findings, problems identified during testing and how they were resolved.

- **Testing methodology.** Describe how the software was tested, which methods, tools and environments were used, and why they were suitable for the software being developed. Methods may include unit, integration, API, functional, system, user scenario, performance, security and compatibility testing, or other appropriate types of testing.

- **Testing of functional requirements.** Describe how compliance with the main functional requirements was verified. Summarise the results, discussing significant, problematic or technically interesting cases in greater detail. It is advisable to identify the problems found, explain their causes and the changes made, and present the results of retesting.

- **Testing of non-functional requirements.** Describe how the relevant non-functional requirements were verified. Testing covers only aspects relevant to the software, such as performance, security, compatibility, reliability or usability.

- **Evaluation of the developed system.** Summarise the development and testing results, assess the extent to which the software meets the specified functional and non-functional requirements, and discuss any remaining limitations and possible improvements.

This section must be 3–5 pages long.

1.2.5. Installation and User Manual

This section must provide the information needed to install, configure, run and use the software system. The student must provide:

- the software system's dependencies on other software products, specifying the systems, services or software components without which it cannot run;
- the specifications of the computer hardware on which the system was developed and tested;
- detailed system installation instructions;
- a description of a typical configuration (if applicable);
- detailed instructions for running the system (particularly for mobile applications, websites or other services based on web technologies);
- the steps for uninstalling the system.

If a particular part is omitted from the user manual, a reasoned explanation must be provided.

This section must be 3–5 pages long.

1.2.6. Conclusions and Recommendations

In this section, the student must present conclusions and recommendations. The conclusions must be substantiated, specific, related to the stated FP aim, consistent with the objectives addressed in the work, and supported by the results obtained. **At least one conclusion must be formulated for each objective.**

The student must evaluate whether the FP aim has been achieved and whether the software developed meets the specified functional and non-functional requirements. If the FP aim has been only partly achieved, some objectives have not been met, or the software does not meet some of the specified requirements, the reasons must be stated, together with an explanation of how this affected the achievement of the FP aim and the software developed.

This section must also recommend ways to extend, improve or apply the software system in practice, indicating possible directions for further development.

If the FP is prepared by a group of students, the conclusions must clearly describe each student's contribution, responsibilities and completed tasks.

This section must be 1–2 pages long.

1.2.7. References

This section lists the sources cited and used in preparing the FP report. Final projects must use APA referencing style, adapted to the institution's requirements for written academic papers and Lithuanian language conventions. Further information: APA Referencing Style – VIKO Library.

Sources may include:

- books;
- periodicals;
- electronic sources, etc.

The reference list must include at least 7 sources.

When software documentation or another electronic resource is used as a source, the specific section or page must be cited, rather than only the documentation homepage or the website homepage. The reference must make it possible to identify the specific source used in the FP.

If a website or software system is the subject of analysis in the FP (e.g. when analysing similar systems), a reference to the system's homepage is sufficient. If statements about the system under analysis are based on information in its documentation or other sources, the specific sources consulted must be cited.

1.2.8. Annexes

Annexes may contain data indirectly related to the FP or material that cannot be presented within the standard text formatting requirements.

Annexes may include:

- written feedback from the company, institution or organisation, if the FP was commissioned;
- examples of results produced by the implemented software system;
- examples of test data for the implemented software system;
- graphic elements, tables, figures or other information mentioned in the FP report but not included in it;
- copies of articles published by the student;
- certificates of participation in conferences;
- other material.

2. ORGANISATION OF FINAL PROJECT PREPARATION, REVIEW AND DEFENCE

The FP is prepared and defended in accordance with the Procedure for the Preparation, Defence and Storage of Final Projects at Vilniaus kolegija, the General Requirements for Written Academic Papers and these Guidelines. The stages and deadlines are set out in the FP Schedule. Students consult their FP supervisor and technical consultant throughout the preparation process. The supervisor must record each student's progress before the FP progress review and before the departmental defence.

Stages of FP preparation, review and defence:

1. Registration of the FP topic with the Department

The FP topic is agreed with the FP supervisor and registered with the Department. When preparing the FP, the student must follow the information provided and approved in the FP Topic Registration Form.

2. FP progress review by the Department Committee

The purpose of the FP progress review is for the Department Committee to assess the work completed to date, including the software developed and the sections of the report prepared.

Before the FP progress review, by the date specified in the Moodle VLE, the student must:

- agree the software under development and the FP report prepared to date with the FP supervisor;
- ensure that the software developed for the FP is stored on *GitHub*;
- upload the FP report prepared to date, as agreed with the supervisor, to the Moodle VLE.

During the FP progress review, the student demonstrates the software developed. Attendance is compulsory.

3. FP defence before the Department Committee

The purpose of the FP defence before the Department Committee is to assess whether the FP meets these Guidelines and the General Requirements for Written Academic Papers.

The student must upload the following to the designated file repository by the specified deadline:

- the completed **FP report** in PDF format, named *Vardas_Pavarde_Grupe*;
- the **source code** of the completed software, named *Vardas_Pavarde_Source_code*;
- **a video** named *Vardas_Pavarde_video*;
- **written feedback from the company, institution or organisation**, if the FP was commissioned.

During the FP defence before the Department Committee, the student presents the FP using slides covering its aim, objectives, results, conclusions and recommendations, and must demonstrate

the working software system. The Department Committee assesses the FP's compliance with these Guidelines and the General Requirements for Written Academic Papers and decides whether to permit the FP report to be uploaded for a similarity check.

4. FP similarity check and formal review

Students authorised at the departmental FP defence to upload their report to the electronic similarity detection system must do so by the deadline in the SCHEDULE FOR FINAL PROJECT PREPARATION, REVIEW AND DEFENCE. The FP supervisor records the conclusion of the similarity check in the supervisor's feedback report and submits it to the Department and the student. Where no instances of plagiarism are identified, the FP may proceed to defence before the Final Project Defence Committee on the Department's recommendation. The designated departmental staff member transfers the FP report to the institution's electronic file repository.

The Department assigns each FP to a reviewer. The review form is provided in Annex 5. Students must familiarise themselves with their reviews no later than one working day before the FP defence.

5. FP defence at a meeting of the Final Project Defence Committee

FPs are defended at a public meeting of the Final Project Defence Committee in accordance with a timetable prepared by the Head of Department and approved by the Dean. The timetable is published on the Faculty website no later than 5 working days before the FP defence. The defence procedures and assessment are set out in the Procedure for the Preparation, Defence and Storage of Final Projects at Vilniaus kolegija.

ANNEX 1. FINAL PROJECT TITLE PAGE IN LITHUANIAN

VILNIAUS KOLEGIJA
ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS

BAIGIAMOJO DARBO TEMA

BAIGIAMASIS DARBAS
BD 6531BX028 PI2XX

STUDENTAS(-Ė) VARDAS PAVARDĖ

20__-__-__

VADOVAS(-Ė) VARDAS PAVARDĖ

20__-__-__

202_

ANNEX 2. FINAL PROJECT TITLE PAGE IN ENGLISH

VILNIAUS KOLEGIJA / HIGHER EDUCATION INSTITUTION
FACULTY OF ELECTRONICS AND INFORMATICS

FINAL PROJECT TITLE

FINAL PROJECT
FP 6531BX028 PI2XX

STUDENT FULL NAME

__/__/20__

ACADEMIC SUPERVISOR FULL NAME

__/__/20__

202_

ANNEX 3. SUMMARY IN LITHUANIAN

SANTRAUKA

Vilniaus kolegija
Elektronikos ir informatikos fakultetas
Programinės įrangos katedra

Studijų programa: Programų sistemos, valstybinis kodas – 6531BX028

Baigiamojo darbo tema: **PAVADINIMAS**
Studentas (-ė): **Vardas Pavardė**
Vadovas (-ė): **Vardas Pavardė**

BD Aprašo apimtis: XXX puslapiai be priedų

BD santrauka turi glaustai ir aiškiai atspindėti darbo turinį, esmę ir pasiektus rezultatus. Joje turi būti nurodytas BD tikslas, trumpai pristatytas darbo turinys ir struktūra bei pateikti pagrindiniai darbo rezultatai. Santraukos apimtis yra 1 puslapis.

Reikšminiai žodžiai: pateikiamos 3-5 pagrindinės sąvokos.

ANNEX 4. SUMMARY IN ENGLISH

SUMMARY

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

Study programme: Software Engineering, state code – 6531BX028

Title of the Final Project: TITLE
Student: First name Surname
Academic supervisor: First name Surname

Length of the FP Report: XXX pages (excluding annexes)

The summary concisely presents the main points of the Final Project and is at least one page long. It must briefly introduce the main issues addressed in each principal part of the report, describe the research methodology, discuss the research results, present the solutions developed in the software part of the project, and state the final conclusions.

The summary must be 1 page long.

Keywords: .....

ANNEX 5. REVIEW FORM

VILNIAUS KOLEGIJA

_____ **FACULTY**

_____ **DEPARTMENT**

FINAL PROJECT REVIEW

Final Project title _____

Student _____

(first name, surname)

Reviewer _____

(first name, surname)

Strengths of the Final Project:

1. Relevance of the problem and analysis of similar systems

2. Task analysis and design

3. Software implementation

4. Results and conclusions

5. Practical significance

6. FP formatting

Weaknesses of the Final Project:

1. Relevance of the problem and analysis of similar systems

2. Task analysis and design

3. Software implementation

4. Results and conclusions

5. Practical significance

6. FP formatting

Questions for the student (no more than 3):

_____

Final Project grade _____

(on a ten-point scale and in words)

Reviewer's workplace and position _____

Reviewer

_____ _____ _____

(date) (signature) (first name, surname)