MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS

SOFTWARE ENGINEERING

Guidelines
for the implementation of the course project
for students of higher education, specialty 121 "Software engineering" of the educational program "Software engineering" of the first (bachelor's) level

Compilers: **Zolotaryova I.O.**

Besedovsky O.M.

D.O. Bondarenko is responsible for the publication .

Kharkiv
HNEU
2024

UDC 004.415 (07)
BBK 32.973-018

Authors: Zolotaryova I.O.
Besedovsky O.M.

Approved at a meeting of the Department of Information Systems.
Protocol No. 7 dated 12/21/2023

Independent electronic text network publication

Methodical recommendations for writing a diploma project for students of higher education, specialty 121 "Software engineering" of the educational program "Software engineering" of the first (bachelor) level [Electronic resource]. Zolotaryova I.O., Besedovskyi O.M. - H.: Ed. HNEU, 2024. - 36 p.

The issue of the organization of course design is outlined, the requirements for the structure of the course project and the design of the explanatory note, methodical recommendations for the development of its structural elements are given.

It is recommended for applicants of the 121 "Software Engineering" specialty of the "Software Engineering" educational program of the first (bachelor's) level.

Introduction

In today's world, software is becoming not only a key component of technological progress, but also a necessary element in various spheres of life, from industry and business to personal devices. Hence, the importance of understanding and mastering software engineering methods and tools becomes clear. For this purpose, methodological recommendations aimed at supporting students in the implementation of this course project have been developed.

Carrying out a course project in software engineering involves a deep understanding of the software development process , including requirements analysis, design, modeling, etc. These recommendations are intended to help students successfully complete the course project.

An important aspect is the systematic approach to software development. This means not only understanding the individual steps in the development process, but also their relationship and impact on the quality of the final product. Methodological recommendations will help students to systematize their knowledge and skills, as well as to perform comprehensive analysis and management of various stages of software development.

A necessary element for the successful implementation of a course project in software engineering is the independent work of applicants with technical literature, modern languages, technologies and environments aimed at improving the work of specialists at all stages of creating a software product.

The course project is an important stage in the preparation of applicants. It is aimed at a deeper understanding and consolidation of theoretical knowledge obtained during the study of the specified discipline, and at improving practical skills in developing project documentation for the software to be created.

These methodological recommendations establish uniform rules and procedures for the execution, registration and protection of the course project. They are intended to help applicants of the 121 "Software Engineering" specialty in completing a course project.

The results of training and competence, which are formed and consolidated by the course project, are defined in the table. 1.

Table 1

Learning outcomes and competences formed by the educational

component

Learning outcomes	Competences that must be mastered by a student of higher education
PH 06	ZK 01, ZK 02, SK 10, SK 11
PH 14	ZK 02, ZK 03, SK 03, SK 04, SK 13

de, RN 06. The ability to choose and use the software creation methodology appropriate to the task.

RN 14. Apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and documentation of software.

ZK 01. Ability to abstract thinking, analysis and synthesis.

ZK 02. Ability to apply knowledge in practical situations.

ZK 03. Ability to communicate in the national language both orally and in writing.

SK 03. Ability to develop architectures, modules and components of software systems.

SC 04. Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards.

SK 10. The ability to accumulate, process and systematize professional knowledge regarding the creation and maintenance of software and the recognition of the importance of lifelong learning.

SK 11. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on relevant models and software development approaches.

SK 13. The ability to reasonably choose and master software development and maintenance tools.

1. Purpose and tasks of course design

The goal of the course project in software engineering is to provide students with the opportunity to learn and practically apply the key principles, methods and tools of software development. In addition, the project is aimed at developing the skills of requirements analysis and software product architecture design.

The course project also aims to teach students to work effectively in a team, using modern methods of collective work and version control systems. As a result of the implementation of the project, the acquirers will be able to develop requirements for software that meets the requirements of the customer and has a stable architecture.

During the development of the course project, the applicant must demonstrate knowledge of:

of the subject area in accordance with the statement of the task;

software development processes;
software development methodology;
on the principles of version control and collective work.
The acquirer must be able to:
carry out the assignment and its analysis;
analyze and formulate requirements;
design system architecture;
work in a team;
manage the project;
communications in the development process;
develop and draw up the necessary documentation.

Work on a course project to a certain extent determines the general theoretical and special training of the applicant and ultimately prepares him for the future performance of a more complex and final stage of the educational process - diploma design. The applicant should consider the work on the course project as a kind of "rehearsal" of the diploma project.

2. Organization of course design

High-quality implementation of the course project requires a clear organization of the applicant's work from the moment of choosing the topic of the project to its defense.

Management of course design is carried out by teachers of the Department of Information Systems. The head of the course project recommends basic literature to the applicant, orients him to the development of the necessary project solutions.

The applicant may be assigned a course project topic from the list of recommended topics (see Appendix A). He is also given the right to independently choose the topic of the project. If the topic is proposed by the applicant, it must be discussed and agreed with the head of the course project.

Materials of the completed course project: the explanatory note in printed and electronic form, the presentation of the report in electronic form, the applicant must submit for review to the supervisor no later than specified period.

The protection of course projects is organized by the department of information systems according to the approved schedule.

During the defense, the applicant briefly reports on the task assigned to him, the project decisions that were made, the results obtained, and answers questions. The report must be accompanied by a demonstration of the presentation, which must contain slides that the applicant determines for a more complete disclosure of the topic.

3. Structure and scope of the course project

The course project consists of an explanatory note and other mandatory materials (schemes, diagrams, graphs, tables, drawings, etc.) developed in accordance with the task.

The volume of the explanatory note is 20-30 printed pages of A4 format (without appendices).

In the table 1 shows the structure of the explanatory note of the course project.

4. Methodological recommendations for the development of structural elements of the explanatory note of the course project

The general requirements for the text of the explanatory note are the logical sequence of the presentation of the material, the clarity and specificity of the presentation of the theoretical and practical results of the work, the essence of the task statement and the purpose of the work, research methods, decisions made, the provenance of the conclusions and the validity of the recommendations. In the text of the explanatory note, it is necessary to adhere to a single terminology. It should not be overloaded with uninformative material, description of well-known data, derivation of formulas, etc. It is necessary to refer to sources of information.

Table 1

The structure of the explanatory note of the course project

Structural elements of the explanatory note	Number of pages
Title page	1
Abstract	1
Content	1
Introduction	1
1 Business analysis of the subject area <name of the subject area>	4 - 10
1.1 Business Requirements	
1.1.1 Background	
1.1.2 Business Opportunity	
1.1.3 Business Objectives and Success Criteria	
1.1.4 Customer or Market Needs	
1.1.5 Business risks (Business Risks)	
1.2 Vision of the Solution (Vision of the Solution)	
1.2.1 Vision Statement	
1.2.2 Major Features	
1.2.3 Assumptions and Dependencies	
1.3 Scope and Limitations	
1.3.1 Scope of initial release of the application (Scope of Initial Release)	
1.3.2 Limitations and Exclusions (Limitations and Exclusions)	
1.4 Business context (Business Context)	
1.4.1 Stakeholder Profiles	
1.4.2 Operating Environment	
2 Specification of requirements for the module (system)	10 - 17
2.1 Analysis of existing solutions (analogues, competitors)	
2.2 Specification of functional requirements	
2.3 Non-functional requirements	
2.4 User interface design (User interfaces)	
2.5 Detailing of non-functional requirements. Logical structure of the data (Detailed Non-Functional Requirements. Logical Structure of the Data)	
Conclusions	1
references	1
Appendices	

The text of the explanatory note should not be written in the first person, preferably the impersonal form (for example, "calculated", "found") throughout the text in the specified case and tense.

When presenting the material, you should not use:

colloquial turns;

slang words and phrases;

different terms for the same concept;

foreign words and terms if there are equivalent words and terms in the Ukrainian language;

abbreviations of words and phrases, in addition to those established by the rules of spelling and regulatory documents.

The title page is the first page of the explanatory note. He contains data that is presented in the following sequence:

information about the performer of the work;

full name of the document;

signatures of responsible persons, including the supervisor;

year of drafting the explanatory note;

An example of a title page is given in Appendix B.

The contents include: introduction; sequentially listed names of all sections, subdivisions and points of the main part of the explanatory note ; conclusions; References; names of appendices and page numbers that contain the beginning of the material.

An example of the content of the explanatory note is given in Appendix B.

In the introduction, it is necessary to identify and formulate a business problem that has arisen in the company, justify the relevance of the project topic to solve this problem based on the module or system under development. Briefly describe the functionality of the module or system. It is necessary to characterize the technical and software platform for the development of the automated module. It is necessary to formulate the goal and tasks of the project, define the object and describe the subject area of the design. It is also necessary to provide information on the design tools that were used in the course project and the possible fields of application of the results obtained in the course project.

The first section "Business analysis of the subject area <name of the subject area>". The purpose of the section is to carry out a detailed analysis of the problem that arose at the management object (company) during business, and to choose ways to solve it.

Subsection 1.1 "Business Requirements".

Business requirements provide the framework and reference for developing all the detailed requirements. Business requirements can be obtained from senior management of the customer or development organization, the executive sponsor, the application manager, the marketing department, or others who have a clear understanding of why the project is being undertaken and the ultimate value it will bring to both the company and to customers

Clause 1.1.1 "Background" should briefly outline the rationale for the new application. Provide a general description of the story or situation that leads to the recognition that this application should be created.

In paragraph 1.1.2 "Business Opportunity" describe the existing market opportunity or business problem being solved. Describe the market in which the commercial application will compete or the environment in which the application will be used. Identify problems that cannot currently be solved without an app and how the app fits into market trends or company strategic directions.

In clause 1.1.3 "Business Objectives and Success Criteria" describe the important business objectives of the application quantitatively and measurably. The value provided to customers will be described in clause 1.1.4, so this clause should focus on the value provided to the business. This may include revenue or cost savings estimates, ROI analysis, or target release dates. Determine how the success of this project will be defined and measured, and describe the factors that are likely to have the greatest impact on achieving this success. Include things directly controlled by the organization as well as external factors. Establish measurable criteria to assess whether business objectives have been achieved.

Example.

Customer:

• high personnel costs;

• the need to increase the number of visitors.

In clause 1.1.4 "Customer or Market Needs", describe the needs of typical customers or market segments, including needs not yet satisfied by the market or existing systems. Describe the problems customers are currently facing that the new app will (or won't) solve, and how customers will use the app. Define the hardware and software environment of the client in which the application must run. Identify at a high level any known critical interface or performance requirements. Avoid including any design or implementation details. Provide requirements as a numbered list so that more detailed user or functional requirements can be traced back to them.

Example.

Clients:

1) Inconvenient ordering procedure

2) Long wait

In Section 1.1.5 Business Risks, summarize the main business risks associated with the development of this application, such as market competition, timing issues, user acceptance, implementation issues, or possible negative impact on business. Assess the severity of the risks and identify any risk mitigation actions that can be taken.

Illustrate the business requirements using a map of goals (Purpose - Direction - Function) (an example is given in Appendix D) and a map of impacts (Impact Mapping) (an example is given in Appendix D).

Subsection 1.2 "Vision of the Solution" defines the long-term vision of the application that must be created to achieve business goals. This vision will provide a context for decision-making throughout the application development lifecycle. The vision should not include detailed functional requirements or project planning information.

Clause 1.2.1 "Vision Statement" describes a concise vision statement that summarizes the purpose and intent of the new application and describes what will happen when the user uses the application. A vision statement should reflect a balanced view that addresses the needs of diverse customers as well as the needs of a growing organization. This may be somewhat idealistic, but it should be based on the realities of existing or anticipated customer markets, enterprise architecture, organizational strategic directions, and cost and resource constraints (Figure 1).

Formula

<App/Project Name> is <app category> for <target audience> which <primary task> using <uniqueness>.

The customer plans to earn/other benefit through [monetization methods].

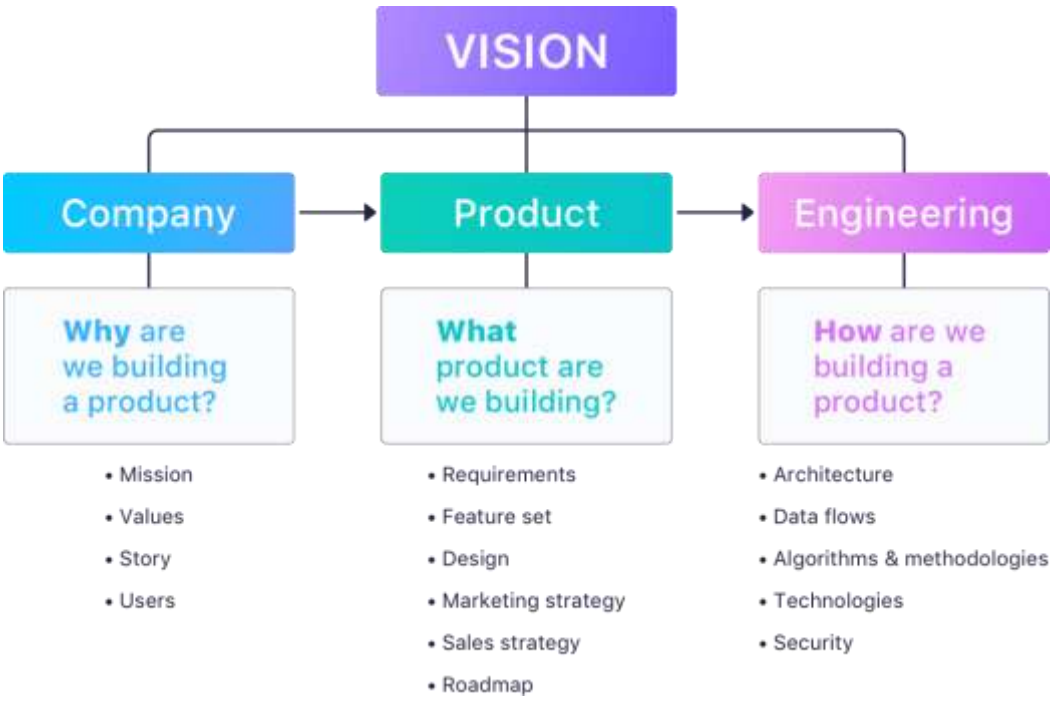

Fig. 1 Structure of the Vision Statement

Clause 1.2.2 "Major Features" includes a numbered list of the main characteristics of the new application, emphasizing those characteristics that distinguish it from previous or competing applications. Specific user requirements and functional requirements may be associated with these features.

Clause 1.2.3 "Assumptions and Dependencies" specifies any assumptions made during the development of the project and the writing of this vision and scope document. Note any major dependencies that the project must rely on for success, such as certain technologies, third-party vendors, development partners, or other business relationships.

Section 1.3 "Scope and Limitations". The scope of the project defines the concept and range of the proposed solution. It is also important to define what will not be included in the application. Clarifying the scope and limitations helps set realistic expectations for multiple stakeholders. It also provides a reference system against which proposed features and requirements changes can be evaluated. Proposed requirements that are beyond the scope of the intended application should be rejected unless they are so beneficial that they should be expanded (with associated changes in budget, schedule, and/or resources).

Clause 1.3.1 "Scope of Initial Release of the Application" describes the planned main functions that will be included in the initial release of the

Application. Describe the benefits that the application should provide to different user communities, and generally describe the application's characteristics and quality characteristics that will enable it to provide these benefits. Avoid including every possible feature that every category of potential customer might ever need. You should focus on those features and characteristics of the application that will provide the most value for the most acceptable development cost for the widest range of users.

Clause 1.3.2 "Limitations and Exclusions" defines any functions or characteristics of the application that the interested party can foresee, but which are not planned to be included in the new application: financial, temporal, legislative, technical.

Section 1.4 "Business Context" summarizes the business issues related to the project, including profiles of the main customer categories, assumptions that went into the project concept, and project management priorities.

It is necessary to submit the glossary in the form of a table (Table 2).

Clause 1.4.1 "Stakeholder Profiles". Stakeholders are individuals, groups, or organizations that are actively involved in the project, are affected by its outcome, or may be affected by its outcome. Stakeholder profiles identify the application's customers and other stakeholders, and indicate their primary interests in the application. Identify business-level customers, target market segments, and different user classes to reduce the likelihood of unexpected requirements that cannot be met due to schedule or scope constraints. For each category of stakeholders, the profile includes the main value or benefits they will get from the application, their likely attitudes towards the application, the main features and characteristics of interest, and any known limitations that need to be addressed.

Table 2

Glossary

Term	Description of the term
1. Basic concepts and categories of the subject area and the project	
2. System users	

An example of a glossary is given in the appendix. J.

Examples of values for stakeholders:

• increase in activity;

• shortened information processing;

• cost savings;

• established business processes;

• automation of tasks that were previously performed manually;

• the ability to perform completely new tasks or functions;

• compliance with applicable standards or regulations;

• improved usability or reduced frustration compared to current applications.

An example of providing profiles of interested parties is given in the appendix. WITH.

In Clause 1.4.2 "Operating Environment" describe the environment in which the system will be used and define the main requirements for availability, reliability, applicability and integrity. This information will significantly affect the definition of the system architecture. Consider questions such as:

• Are the users geographically distributed or close together? How many time zones are they in?

• When do users in different locations need access to the system?

• Where is the data generated and used? How far are these places from each other? Need to combine data from multiple locations?

• Is there a specific maximum response time known for accessing data that may be stored remotely?

• Can users tolerate service interruptions or is continuous access to the system critical to their business operations?

• What access security controls and data protection requirements are required?

The purpose of the second section "Specification of requirements for the module (system)" is to develop and detail the requirements specification for the module (system) being developed.

A Module (System) Requirements Specification (SRS) should contain sufficient information to enable developers to complete the creation of the described software. It not only describes the software being developed, but also the purpose it will serve: what the software should do and how it should work.

An SRS document typically includes the following elements:

- the purpose of the software being developed;
- general description of the software;
- the functionality of the software or what it is supposed to do;
- applicability of software in a production situation;
- non-functional requirements;
- external interfaces or how the software will interact with hardware or other software to which it must connect;
- design limitations or limitations of the environment in which the software will operate.

In subsection 2.1 "Analysis of existing solutions (analogues, competitors)" a brief description of analogues / competitors (at least 3-5) should be provided.

The comparative analysis should be carried out according to the criteria (Table 3).

Table 3

Comparative analysis of existing solutions by criteria

	Analog 1	Analog 2	…	Analog 5
Criterion 1				
Criterion 2				
…				
Criterion 6				

The comparative analysis should end with a conclusion about advantages / disadvantages, as well as the need to develop your own solution.

Subsection 2.2 "Specification of functional requirements". Functional requirements for the system under development are software requirements

that describe the internal workings of the system, its behavior: data calculation, data manipulation, data processing, and other specific functions that the system must perform. Unlike non-functional requirements, which define what a system should be, functional requirements define what the system should do.

Development of use cases includes:

- diagram of use cases;
- specification of use cases.

A use case diagram represents the functionality that will be implemented in a software application. A use case can be considered as a function implemented by the system. However, any feature should have value and provide an end result for the end user of the application or service. Therefore, when specifying the use case, among all the functionality of the system, only that functionality is singled out, which:

- useful to a specific end user **;**
- allows the user to receive specific finished results.

Specification of use cases.

Each use case should have a description. The project should include a description of the use cases that implement the main functionality (usually, except for keeping directories) in the form of a table (Table 4).

Table 4

Specification of use cases

The name of the use case	
Brief description	
The characters	
Prerequisite	
Main flow of events (script)	1. 2. ...
Alternative event flow	1. 2. ...
Posthumous	
Exclusion	

Each use case should be accompanied by an Activity diagram.

Subsection 2.3 "Non-functional requirements" should contain:

- performance requirements;
- reliability requirements;
- security requirements;
- software application quality attributes;
- requirements of internationalization;
- database requirements.

In subsection 2.4 "Designing User Interfaces (User Interfaces)", you need to develop a graphical user interface with the application in the form of a wireframe or mockup, which is used to obtain the approval of interested parties for the proposed concept.

This section should describe:

1. Logic between users and software. Attention should be paid to detailing everything from screen images to app layouts, menu content to reports, and more.

2. Ways to optimize the interface. Important guidelines and limitations for user interface optimization should be listed.

Next, any communication interfaces with other systems or devices, such as local networks, remote serial devices, etc., should be described. The described project should support all types of web browsers.

To visualize the software application being developed, program frameworks, layouts, and prototypes should be added to SRS (Fig. 2).

• A wireframe is an image of a structure with a low level of detail. This is usually a black and white layout of the application page. It outlines the location of the main elements: buttons, images, texts. It does not perform any real site functions. The results of interactions, clicks, animations must be described additionally in the comments. A framework is used to determine where and what content will be placed.

• A mockup is a detailed static representation of a design. It includes colors, images, typography. A layout is more like a nice image of an app or website. A layout is used to create a project style, present and coordinate visual details with the customer.

• Prototype is a simulation of user interaction with the application interface. This is an interactive version of the framework. The prototype is used to illustrate the arrangement of blocks and buttons and to conduct usability testing.

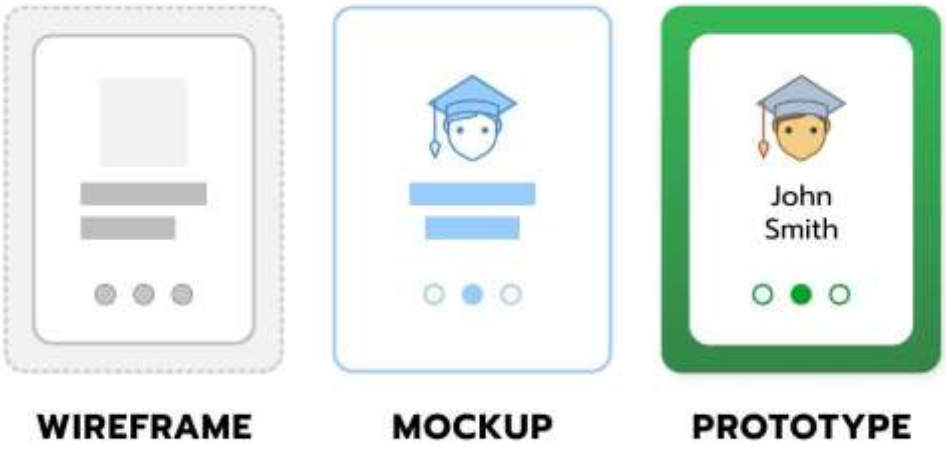

Fig. 2 Examples of framework, layout, prototype

The purpose of interface design is to obtain early user response to the proposed system concept using non-cost means. Marvel, Pencil Project, Use Your Interface, UX Myths, Mobile Patterns, Good UI, Microsoft Visio, Microsoft PowerPoint, etc. are used as tools.

Subsection 2.5 "Detailing of non-functional requirements. Logical structure of the data (Detailed Non-Functional Requirements. Logical Structure of the Data)". Designing an application's data structure includes database design and class diagram design. These two approaches are necessary because the application needs to store data.

Therefore, a discussion of system design will include an entity-relationship (ER) diagram and a class diagram.

After determining the requirements for the application, it is necessary to determine the various entities participating in it. First, it is a logical ER diagram - an "entity-relationship" diagram, in which the entities must be normalized to establish the proper relationships. A detailed description of the database tables for the system being developed should also be made.

Next, you should provide the UML class diagram (Class Diagram) that implements the basic business logic of the software system, and its description.

Examples of a logical ER diagram, a description of database tables, and a UML class diagram are given in the appendix. K.

Conclusions for the work are a summary of the results of the entire work. This part is of particular importance, since the final results of the work should be given here.

The explanatory note should contain conclusions on each stage of the completed project and on the work as a whole, which must be correlated with the goal and task of the course design.

It is necessary to note the practical value of the work results, to give recommendations for further improvement of the design object. Noting the practical value of the obtained results, it is important to outline the degree of their readiness for use, the scope of use.

The list of used sources should contain information about the literary sources used in the development of the project.

The bibliography is a register of used sources on the topic of the project in the broadest possible sense. Therefore, one should not limit oneself to only cited literature. The list should include all materials that were read, reviewed, analyzed while working on the course project and related to its topic. It is desirable to identify the sources as fully as possible, remembering that the

bibliographic list for the project is a summary of the study of the problem and a prerequisite for further scientific research.

The list of used sources must contain at least 30 sources. It is presented in the original language, placed in alphabetical order of the first authors' surnames or titles, and numbered in ascending order. Numbering is continuous.

The works of one author are arranged alphabetically by title or in the chronology of their writing. The alphabetical list is arranged alphabetically in the following sequence:

literature of the combined Cyrillic alphabet, for sources in languages with Cyrillic graphics (Russian, Ukrainian, Bulgarian, etc.);

literature in the Latin alphabet,

electronic resources in the same sequence as printed editions (first in Cyrillic and then in Latin).

The list of literary sources must contain the surname and initials of the author, the full name of the source, the city of publishing, the publishing house and the year of publication, the number of pages or links to pages, etc. The total volume of the book in pages is indicated if the reference is made to it in its entirety, the pages (from... to) are marked if the reference refers to a separate part of the literary source.

Appendices contain material that is necessary for the completeness of the explanatory note, but cannot be consistently placed in its main part due to the large volume or for other reasons.

Illustrations (diagrams of business processes, dialogue scripts, etc.), tables, intermediate mathematical proofs, formulas and calculations, auxiliary text, etc. can be made in the form of applications.

5. Requirements for the design of the explanatory note of the course project

Of great importance when working on a course project is its design, which is subject to certain requirements.

When designing the text of the explanatory note, you should be guided by the Methodological recommendations for the design of reports, course projects and diploma theses (projects) [1] .

Recommended Books

The main one

1. Methodical recommendations for the preparation of reports, course projects and diploma theses (projects) for students of specialties 121 "Software engineering", 122 "Computer science", 126 "Information systems and technologies" [Electronic resource] / comp. I. O. Ushakova, G. O. Plekhanova, O. M. Besedovskyi. – Kharkiv: KHNEU named after S. Kuznetsia, 2021. - 45 p. - Access mode: http://repository.hneu.edu.ua/handle/123456789/27413

Additional

2. Software engineering: the working program of the educational discipline for higher education applicants, specialty 121 "Software engineering" of the educational program "Software engineering" of the first (bachelor's) level: [Electronic edition] / comp. I.O. Zolotaryova. – Kharkiv: KHNEU named after S. Kuznetsia, 2024. - 13 p. - Access mode: http://repository.hneu.edu.ua/handle/123456789/32174

3. Borodkina I.L., Borozhkin G.O. Software engineering. – Center for Educational Literature, 2020. – 204 p.

Internet resources

4. Software Requirements Specification document with example. [Electronic resource]. – Access mode: https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

5. How to Write a Software Requirements Specification (SRS Document). [Electronic resource]. – Access mode: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

6. Guide to Software Requirements Specification (SRS) Document 2022. [Electronic resource]. – Access mode: https://youteam.io/blog/guide-to-software-requirements-specification-srs-document/

7. The Airline Ticket Booking System Example. [Electronic resource]. – Access mode: https://flylib.com/books/en/2.96.1.35/1/

8. Personal educational system "Course project "Software engineering" (6.121.010)" [Electronic resource]. - Access mode: https://pns.hneu.edu.ua/course/view.php?id=6597

Appendices

Appendix A

List of recommended topics of course projects

Smart cities

1. Street lighting monitoring and optimization system:
   - Development of software for collecting data on street lighting, motion sensors, and forecasting lighting needs.
   - Creation of algorithms for dynamic lighting control in order to save energy and improve visibility.
2. System of monitoring and management of parking places:
   - Development of a mobile application for finding free parking spaces in real time.
   - Creation of a system of sensors to track the occupancy of parking spaces.
3. Air pollution monitoring and forecasting system:
   - Development of software for collecting air quality data from sensors located throughout the city.
   - Creation of algorithms for forecasting air pollution and alerting residents.
4. Municipal waste monitoring and management system:
   - Development of a mobile application for reporting overflowing garbage cans and ordering garbage removal.
   - Creation of a system of sensors to monitor the filling level of garbage cans.

Bomb shelter map

5. Mobile application with a map of bomb shelters:
   - Development of a mobile application with an interactive map of bomb shelters, including information about their location, capacity, availability and other characteristics.
   - Adding the function of finding the nearest bomb shelter and laying a route to it.

Continuation of Appendix A

6. Website with a map of bomb shelters:
    - Creation of a website with a map of bomb shelters containing information on their location, capacity, accessibility and other characteristics.
    - Adding the function of searching for bomb shelters by address, district or other criteria.
7. Online registration system for receiving social services:
    - Software development for online registration of people who need social services.
    - Creation of algorithms for automated distribution of resources and maintaining contact with people who need help.
8. System of monitoring and support of the elderly:
    - Development of a mobile application for monitoring the state of health and vital activities of the elderly.
    - Creating a sensor system to track health indicators and report emergencies.
9. Online learning system for people with disabilities:
    - Development of a web platform with online courses and interactive materials for people with disabilities.
    - Creating an adaptive interface and accessible content for people with different types of disabilities.
10. Online platform system for volunteers:
    - Development of a web platform to connect volunteers with people who need help
    - Creation of algorithms for selecting volunteers based on their skills and experience.

Other topics

11. System of monitoring and management of water resources of the city
12. Early warning system for natural disasters
13. Online voting system for participation in local self-government
14. A system of monitoring and improving the accessibility of public space

End of Appendix A

15. An online platform system for sharing knowledge and experience
16. An online platform system for fundraising for social projects
17. A system of monitoring and improving the quality of life in the community

Appendix B

Cover page design sample

KHARKIV NATIONAL ECONOMIC UNIVERSITY
NAMED AFTER SEMEN KUZNETS
FACULTY OF INFORMATION TECHNOLOGIES
DEPARTMENT OF INFORMATION SYSTEMS

explanatory note

to the course project in "Software Engineering"
on the topic: "Designing the module "Name of the subject area""

Winner(s) of the 3rd year of the
6.04.121.010.21.01 group
Specialties 121 "Software engineering
software"
first (bachelor) level
Varenyk G.V.
Head: Professor of the IS Department, Doctor of Economics, Associate Professor,
Zolotaryova I.O.

National scale _____

Number of points: _____Grade: ECTS _____

Members of the commission

_____ _____
(signature) (surname and initials)

_____ _____
(signature) (surname and initials)

Kharkiv - 202_ year

Appendix B

Sample design of the content of the explanatory note

CONTENT

Appendix G

An example of a goal map

Fig. D.1 Analysis of business requirements in the form of a goal map

Appendix D

An example of an influence map

Table D.1

Analysis of business requirements in the form of an impact map

Why (goal)	**Who (The characters)**	**As (Influence, action)**	**What (tasks, requirements)**
Reduce the cost of maintaining waiters by X	Manager	Manage the electronic menu	Manage the database of available dishes
			Check and confirm orders
	Waiter	Reduce the amount of work performed	Be engaged only in serving food and receiving payment
	Customer	Do not contact the waiters to order	Use the electronic menu
Increase the number of modern customers by N	PR manager	Conduct a PR campaign for the introduction of an electronic menu	Send SMS and email to regular customers
			Buy advertising in social networks

Appendix G

Glossary example

Table Z.1

Glossary

Term	Description of the term
1. Basic concepts and categories of the subject area and the project	
Electronic menu	A web application that has access to a database of products available for ordering and provides the functionality for the user to complete the order.
Order	List of dishes chosen by the customer.
2. Users of the application	
Customer	A person who has access to the system to fulfill an order.
Manager	A person who has access to a system with special rights to manage information.

Appendix C

Example of stakeholder profiles

Table C.1

Stakeholder Profiles

User classes	User characteristics
Manager	A manager is a user who manages the order database. *List of possibilities:* 1. Adding information about new dishes. 2. Editing and deleting information about existing dishes. 3. Order confirmation.
Waiter	A waiter is a user who takes the finished order to the customer. *List of features:* 1. Obtaining information about the location of the customer 2. Delivery of ready meals to the customer 3. Acceptance of payment for the order
Customer	A customer is a user who places an order for certain dishes. *List of possibilities:* 1. Acquaintance with the list of dishes available for ordering 2. Order execution 3. Payment for the order

Appendix K

An example of the design of the database design unit

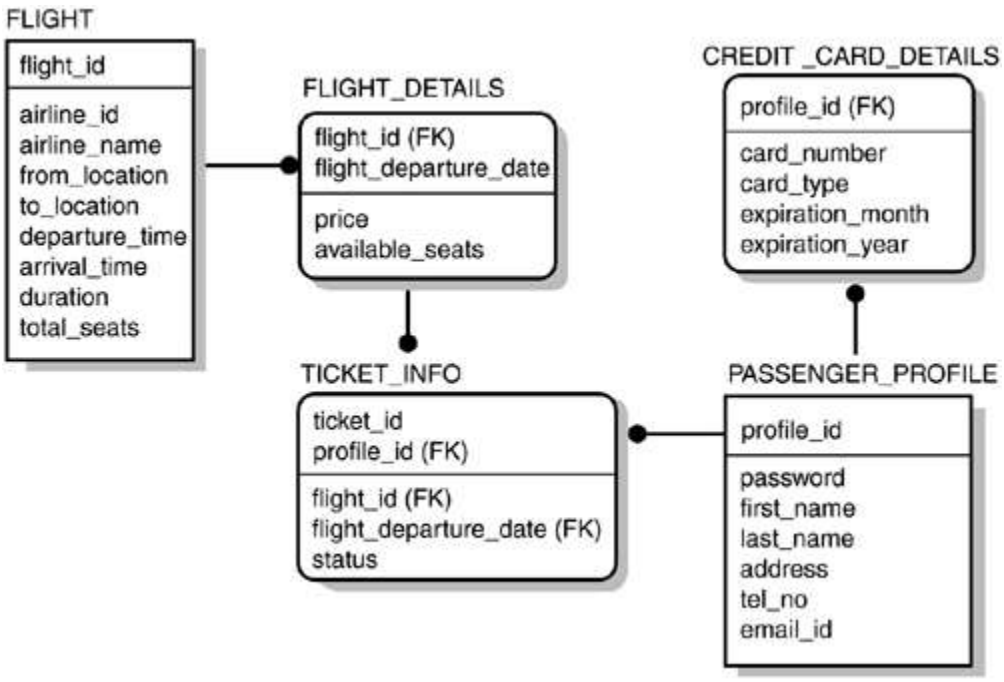

Fig. K.1 Logical database model

Table K.1

Structure of the "FLIGHT" table

Column Name	Data Type	Remarks
flight_id	NUMBER	Represents the unique identifier for a flight of an airline.
airline_id	NUMBER	Represents the unique identifier for an airline.
airline_name	VARCHAR(20)	Represents the unique identifier for an airline.
from_location	VARCHAR(20)	Represents the location from which the flight begins.
to_location	VARCHAR(20)	Represents the destination of the flight.
departure_time	DATETIME	Represents the time the flight departs every day.
arrival_time	DATETIME	Represents the time the flight arrives every day.
duration	NUMBER	Represents the time taken for the flight.
total_seats	NUMBER	Represents the number of seats on this flight.

Table K.2

Structure of the "FLIGHT_DETAILS" table

Column Name	Data Type	Remarks
flight_id	NUMBER	Represents the unique identifier for a flight. This is a foreign key.
flight_departure_date	DATETIME	Represents the departure date for the flight.
price	NUMBER	Represents the price of the flight on a specific date.
available_seats	NUMBER	Represents the seats available on a specific date for a flight.

Table K.3

Structure of the "PASSENGER_PROFILE" table

Column Name	Data Type	Remarks
profile_id	NUMBER	Represents the unique identifier for a passenger. This is a primary key.
password	VARCHAR(20)	Represents the password used by the passenger to authenticate with the application.
first_name	VARCHAR(20)	Represents the first name of the passenger.
last_name	VARCHAR(20)	Represents the last name of the passenger.
address	VARCHAR(20)	Represents the address of the passenger.
tel_no	VARCHAR(20)	Represents the telephone number of the passenger.
email_id	VARCHAR(20)	Represents the e-mail ID of the passenger.

Table K.4

The structure of the "TICKET_INFO" table

Column Name	Data Type	Remarks
ticket_id	NUMBER	Represents the unique identifier for a ticket.
profile_id	VARCHAR(8)	Represents the unique identifier for a passenger. This is a foreign key.
flight_id	NUMBER	Represents the flight number of the flight booked by the passenger. This is a foreign key.
flight_departure_date	DATETIME	Represents the departure date of the flight booked by the passenger.
status	VARCHAR(10)	Represents the status of the ticket purchased by a passenger.

Table K.5

Structure of the "CREDIT_CARD_DETAILS" table

Column Name	Data Type	Remarks
profile_id	NUMBER	Represents the unique identifier for a passenger. This is a foreign key.
card_number	VARCHAR(20)	Represents the credit card number used by the passenger to purchase tickets.
card_type	VARCHAR(20)	Represents the credit card type used by the passenger.
expiration_month	VARCHAR(2)	Represents the expiration month of the credit card of the passenger.

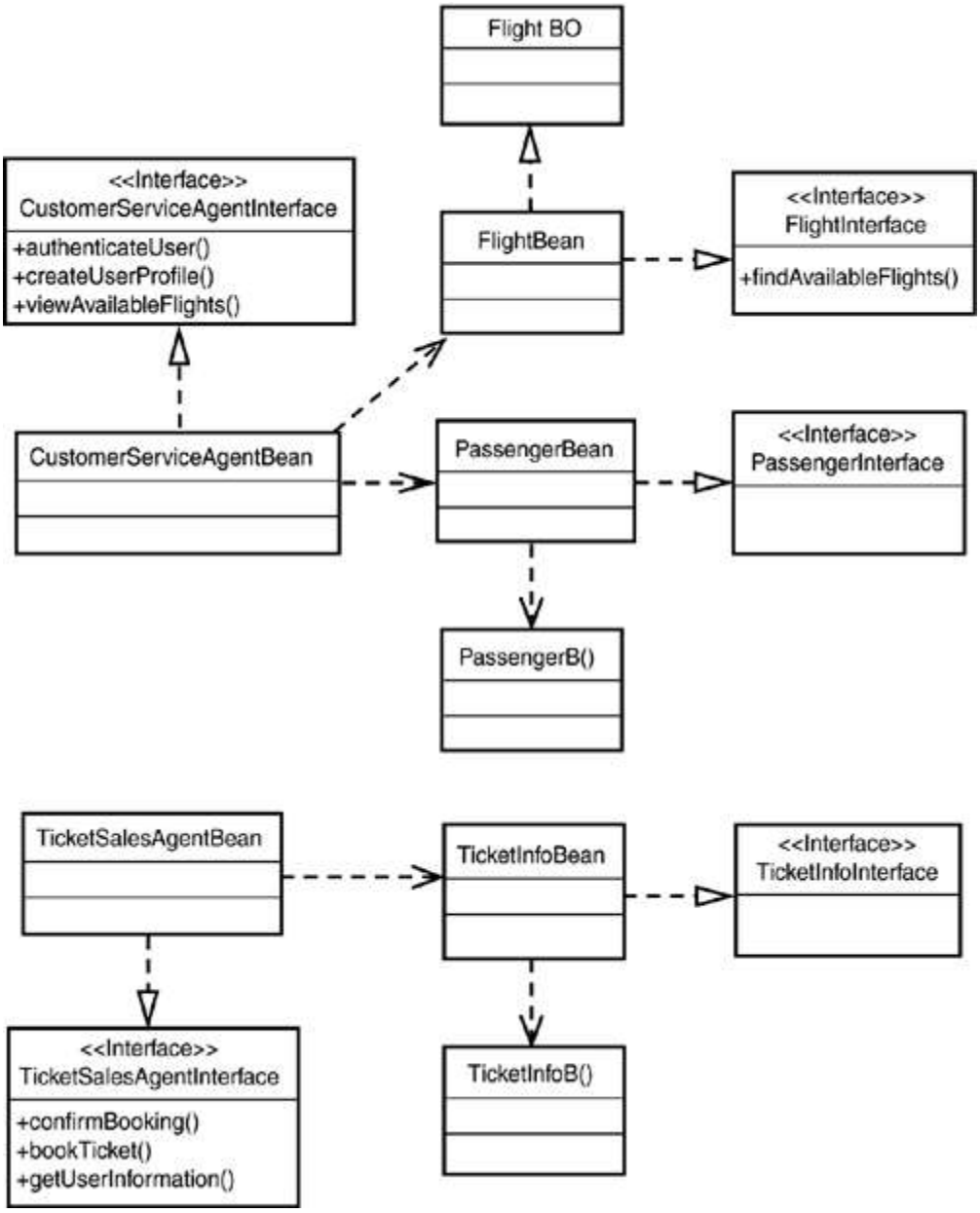

Fig. K.2 UML class diagram

Content

EDUCATIONAL EDITION

SOFTWARE ENGINEERING

Methodological recommendations for the implementation of the course project for students of higher education specialty 121 "Software engineering" of the educational program "Software Engineering" of the first (bachelor's) level

Compilers: Iryna Oleksandrivna Zolotaryova
Oleksiy Mykolayovych Besedovskyi

D.O. Bondarenko is responsible for the release.

Editor
Corrector

Plan 2024. Pos. No. 7 9 EV .
Sub for printing Format 60 x 90 1/16. MultiCopy paper. Riso print.
Artificial intelligence sheet Region-issue sheet Circulation approx. Deputy No
Publisher and manufacturer — publishing house of Khneu, 61166, Kharkiv, Nauky avenue, 9-A
Certificate of entry of the subject of the publishing business into the State Register, Dk No. 4853 dated February 20, 2015.