Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of Radio Engineering Systems

[RE-319] ARCHITECTURE OF CROSS-PLATFORM SOFTWARE SOLUTIONS

Curriculum of the academic discipline (Syllabus)

Course details

Level of higher education	First (bachelor's)
Field of knowledge	17 - Electronics, Automation, and Electronic Communications
Specialization	172 - Electronic Communications and Radio Engineering
Educational program	All educational programs
Discipline status	Elective (F-catalog)
Form of higher education	Full-time
Year of training, semester	Available for selection starting from the 3rd year, fall semester
Scope of the discipline	4 credits (Lectures 18 hours, Practical classes 36 hours, Laboratory work 36 hours, Independent work 66 hours)
Semester Control/control measures	Credit
Class schedule	https://schedule.kpi.ua
Language of instruction	Ukrainian
Information about the course coordinator/teacher s	Lecturer: P. Yu. Katyn Lab: Katin P. Yu.
Course location	

Curriculum

1. Description of the academic discipline, its purpose, subject matter, and learning outcomes

The purpose of studying the discipline (credit module) "Architecture of Cross-Platform Software Solutions" is to develop students' ability to independently develop software that implements the software component of radio engineering systems. The ASP.NET Core MVC framework is used as a basis.

The subject of the academic discipline (module) is the architecture of software that can potentially be implemented in radio engineering systems. Studying the discipline (credit module) allows students to:

- develop, debug and maintain web applications based on the ASP.NET Core MVC architecture framework;
- develop software architecture based on the ASP.NET Core MVC framework;
- apply fundamental knowledge of software architecture for professional development of web applications.

Program outcomes of the discipline (module):

- Recognize the main architectural styles of web application software systems and describe the system architecture;
- Develop medium-complexity software systems that meet the requirements for web applications; • Design and develop source code for web applications based on typical framework architectures;
- Know and apply relevant architectural concepts, methods, system and object-oriented analysis for web applications during software development;
- Apply effective approaches to software architecture in practice;
- Know and be able to use the fundamental architecture of modern software in web development technology;
- Know typical software architecture solutions and be able to apply them to web application development;
- Know the principles of using the latest technologies in web development and trends in their development;
- Be able to identify, analyze, and document the software architecture of web applications.

2. Prerequisites and post-requisites of the discipline (place in the structural-logical scheme of training under the relevant educational program)

Successful completion of the discipline (credit module) precedes the study of the discipline "Informatics" in the curriculum for bachelor's degree programs in specialty 172.

The theoretical knowledge and practical skills acquired during the course (credit module) are necessary for the development of a bachelor's project and a master's training plan in specialty 172.

3. Course content

Topic 1. Fundamentals of software and hardware components of radio engineering systems

Topic 2. Object-oriented programming (OOP) techniques at the architectural level Topic 3.

Fundamentals of programming architecture in the object-oriented paradigm.

Topic 4. Fundamentals of ASP.NET Core MVC web application architecture Topic

5. Stages of creating and debugging an ASP.NET Core MVC project Topic 6.

Databases in web application architecture

Topic 7. Elements of JavaScript technology in web application architecture

Topic 8. Architecture of a practical web application

Topic 9. Creating software and hardware for modern radio systems with web application elements Modular

test #1, CREDIT

4. Teaching materials and resources

Basic literature

1. Software infrastructure for WEB applications. Laboratory workshop [Electronic resource]: textbook for students majoring in 121 "Software Engineering" and 126 "Information Systems and Technologies" / M. M. Bukasov, D. O. Galushko, P. Yu. Katin, Ya. V. Khitsko; Igor Sikorsky KPI. – Electronic text data (1 file: 780.7 KB). – Kyiv: Igor Sikorsky KPI, 2023. – 53 p. – Title from screen.https://ela.kpi.ua/handle/123456789/53028?mode=full

2. Ostapchenko, K. B. Databases. Computer workshop [Electronic resource]: textbook for bachelor's degree students majoring in 126 Information Systems and Technologies / K. B. Ostapchenko; Igor Sikorsky Kyiv Polytechnic Institute. – Electronic text data (1 file: 1.39 MB). – Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2022. – 151 p. – Title from screen.

3. Konovalenko I.V. .NET Platform and C# 8.0 Programming Language: Textbook / Konovalenko I.V., Marushchak P.O. – Ternopil: FOP Palianytsia V. A., 2020 – 320 p. https://elartu.tntu.edu.ua/bitstream/lib/32825/1/Konovalenko%20I.%20.NET-C%23.pdf

Additional reading.

1. Adam Freeman. Pro ASP.NET Core MVC 2 London, UK.

2. Karli Watson. Jacob Vibe Hammer. John D. Reid. Morgan Skinner. Daniel Kemper. Christian Nagel. BEGINNING Visual C#® 2012 Programming. Published by John Wiley & Sons, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256. www.wiley.com.

3. Object-oriented Programming in C# for C and Java programmers. February 2010. Kurt Nørmark © Department of Computer Science, Aalborg University, Denmark. URL: http://www.cs.aau.dk/~normark/oop-csharp/html/notes/theme-index.html.

4. Paul Michaels. Software Architecture by Example: Using C# and .NET. Derbyshire, UK. URL: doi.org/10.1007/978-1-4842-7990-8

5. https://www.dofactory.com/net/design-patterns

6. http://www.orangepi.org

Educational content

5. Methodology for mastering the academic discipline (educational component)

No.	Type of classes	Description of class content
Topic 1. Fundamentals of software and hardware components of radio engineering systems		
1	Lecture 1. Fundamentals of hardware solutions for building programmable elements of radio engineering systems.	Hardware components of modern microcomputers as a potential component of radio engineering systems and complexes. Microcomputer operating systems. Software technologies for implementing the software architecture of radio engineering systems. Examples of Orange Pi microcomputers.

2	Lecture 2. Fundamentals of software technologies for building programmable elements of radio engineering systems	Software for mini-computers, operating systems, and technologies. ASP.NET Core MVC for mini-computers. Web applications as the basis of software technologies for radio engineering systems.
3	Computer workshop 1.	Installation and testing of the development environment and debugging ASP.NET Core MVC for Orange Pi
Topic 2. Object-oriented programming (OOP) techniques at the architectural level.		
4	Lecture 3. Fundamentals of object-oriented programming	Object-oriented programming techniques. Classes, constructors, destructors, static class members, fields, methods, properties. Software interfaces, encapsulation, inheritance, polymorphism. Software solutions for demonstrating materials.
5	Lecture 4. Patterns and basic principles of object-oriented programming	The culture of object-oriented programming. SOLID principles. Programming templates. Examples of using programming patterns to create software architecture. Relationships between objects in the OOP paradigm. Containers and patterns. Operator overloading. Software solutions for demonstrating materials.
6	Computer workshop 2.	Practical class to reinforce the OOP paradigm programming in C#.
Topic 3. Fundamentals of programming architecture in the object-oriented paradigm.		
7	Lecture 5. Basic elements of the C# programming language technology for web development	Creating a web application project in MVC architecture. Debugging and containerization of a web application. Software solution for demonstrating OOP principles in ASP.NET Core MVC.
8	Lecture 6. Documenting software application architecture.	Documenting software architecture in an object-oriented paradigm. Basics of the UML programming language. Interfaces, inheritance, content, inclusion. Creating UML class diagrams, sequence diagrams, component diagrams, deployment diagrams based on the obtained software solutions.
9	Computer workshop 3.	Practical class on consolidating the OOP programming paradigm in C# with documentation in the form of UML.
Topic 4. Fundamentals of ASP.NET Core MVC web application architecture.		
10	Lecture 7. Types of web application architecture and their implementation in ASP.NET Core MVC technology	Types of web application architecture in the form of a typical MVC pattern. Understanding the model, controller, and view. Implementation of the MVC pattern and its operation in ASP.NET Core MVC technology.
11	Lecture 8. Development and testing of a controller in ASP.NET Core MVC web application	Creating the basis of a web application in ASP.NET Core MVC and exploring related C# programming technologies C# programming language.
12	Computer workshop 4.	Creating web application elements based on ASP.NET.
Midterm Test		
Topic 5. Stages of creating and debugging an ASP.NET Core MVC project.		

13	Lecture 9. Creating a web application prototype in ASP.NET Core MVC Project	Improving the architecture of a web application based on ASP.NET. Creating and adding dynamic output. Improving the data model. Creating a linked method with activity.
14	Lecture 10. Basics of information display technology in ASP.NET Core MVC Project	Retrieving and dynamically displaying information from a database on a web interface. Basics of web application validation. Adding styles and display frameworks to web interfaces. Architectural solutions and patterns: Smart UI Pattern, Model-View Architecture, Three-Tier Architecture.
15	Computer workshop 5.	Improving web application architecture based on ASP.NET.
Topic 6. Databases in web application architecture		
16	Lecture 11. Interaction between the database and the web application in the context of the model and business logic	Architectural component of a web application based on the model and business logic. Technologies for interaction between the database and the web application user application. Documenting the solution.
17	Lecture 12. Basic technologies for working with C# databases	Architectural component of a web application based on ASP.NET technologies. Technologies for database and user interaction with the web application based on the code first principle.
18	Computer workshop 6.	Creating a web application project based on ASP.NET with a database example.
Topic 7. Elements of front-end technology in web application architecture		
19	Lecture 13. Fundamentals of HTML5 and CSS style sheets in web application architecture	Basics of HTML5 and CSS technology. Tags, attributes, and styles. Templates for modern web interface solutions. Creating professional display systems using modern technologies. Advantages and disadvantages of JavaScript when creating elements web application display elements.
20	Lecture 14. Razor technology	Razor technology, disadvantages and advantages. Examples of using technologies when creating web application display elements.
21	Computer workshop 7.	Improving web application architecture Razor technology.
Topic 8. Architecture of a practical web application		
22	Lecture 15. Forming and documenting the architecture of a professional ASP.NET web application Core.	Documenting a complex MVC web application. Creating modular tests for a web application. Organization of the domain model.
23	Lecture 16. Version control system for collaborative web application development	Git introduction to technology. Git and GitHub professional work with the version control system. Git, branching of project options.
24	Computer workshop 8.	Documenting and testing an ASP.NET project.
Topic 9. Creating software and hardware for modern radio systems with web application elements applications		
25	Lecture 17. Types of architecture and technologies of modern web applications	Monolithic architecture. Microservice architecture. Pattern-based architecture (Patterns (supplemental)). Using cloud technologies to form application architecture applications.
26	Lecture 18. Trends in the development of web application architecture	Final review of the material. Conclusions. Prospects for the development of web programming technologies and architecture. Creating prototypes of web solutions for work.
27	Computer workshop 9.	Creating a project and deploying it on Azure.
Exam		

6. Independent work

The discipline is based on independent preparation for classroom sessions on theoretical and practical topics.

No	Area of independent preparation	Number of hours	Literature
1	Preparation for lecture 1	1	1
2	Preparation for practical class 1	1.5	1
3	Preparation for lecture 2	1	1
4	Preparation for lecture 3	1	1
5	Preparation for practical class 2	1.5	1
6	Preparation for lecture 4	1	1
7	Preparation for Lecture 5	1	1-5
8	Preparation for practical class 3 (part 1)	1.5	1
9	Preparation for lecture 6	1	1
10	Preparation for lecture 7	1	1
11	Preparation for practical class 3 (part 2)	1.5	1
1	Preparation for lecture 8	1	1
13	Preparation for lecture 9	1	1
14	Preparation for practical class 4 (part 1)	1.5	1
1	Preparation for lecture 10	1	1
16	Preparation for lecture 11	1	1
17	Preparation for practical lesson 4 (part 2)	1.5	
18	Preparation for lecture 12	1	
19	Preparation for lecture 13	1	
20	Preparation for practical class 5	1.5	
21	Preparation for lecture 14	1	1
2	Preparation for lecture 15	1	1
23	Preparation for practical class 6 (part 1)	1.5	1
24	Preparation for lecture 16	1	1
25	Preparation for lecture 17	1	1
26	Preparation for practical class 6 (part 2)	1.5	1
27	Preparation for the midterm test	4	1-5
28	Preparation for the exam	30	1
29	Getting started with MVS	2	1
30	Install MVS	1	1
31	Essential C# Features for web developing. Stage 1.	1	1
32	Essential C# Features for web developing. Stage 1.	1	1
33	Essential C# Features for web developing. Stage 2.	1.5	1
34	Essential C# Features for web developing. Stage 2.	1	1
35	Essential C# Features for web developing. Stage 3.	1	1
36	Essential C# Features for web developing. Stage 3.	2	1
37	Essential C# Features for web developing. Stage 4.	2	1
38	Essential C# Features for web developing. Stage 4.	1	1
39	Entity Framework Core Tools Package	1	1
40	Lambda Expressions	2	1

Policy and control

7. Academic Discipline Policy (Educational Component)

Attendance at lectures is mandatory. In exceptional circumstances, attendance requirements and other aspects of the policy may be subject to change.

Attendance at computer lab classes may be sporadic and, if necessary, for consultation/defense of computer lab work.

Rules of conduct in class: be active, respect others, turn off phones. Follow the academic integrity policy.

Rules for defending computer lab work: work must be done in accordance with the tasks set and according to the option.

The rules for awarding incentive and penalty points are as follows.

Bonus points are awarded for accurate and complete answers in surveys based on lecture materials (maximum number of points for a survey is 3 points).

8. Types of assessment and the learning outcomes assessment rating system (LOAS)

During the semester, students complete 5 computer workshops. The maximum number of points for each computer workshop is 10 points.

Points are awarded for:

- quality of the computer workshop: 0-5 points;
- answers during the defense of the computer workshop: 0-3 points; •

timely submission of work for defense: 0-2 points.

Criteria for assessing the quality of performance:

- 5 points – the work is done well and in full;
- 4 points – the work is done well, in full, but has some flaws; • 3

points – the work is complete, but contains minor errors;

- 2 points – the work is completed in full, but contains significant errors; • 0

points – the work is not completed in full.

Answer evaluation criteria:

- 3 points – the answer is complete and well-reasoned;
- 2 points – the answer is correct, but has shortcomings or minor errors; • 1

point – the answer contains significant errors;

- 0 points – no answer or incorrect answer.

Criteria for assessing the timeliness of submitting work for defense:

- 2 points – the work is submitted for defense no later than the specified deadline; • 0

points – the work was submitted for defense after the specified deadline.

- Maximum number of points for completing and defending computer practicals: • 10

points × 5 computer workshops = 50 points.

During the semester, quizzes on the topic of the current lesson are held during lectures. Maximum number of points for all quizzes: 3 points. The number of quizzes on the topic of the current lesson for one student is unlimited.

The module test consists of 3 theoretical and 2 practical questions. Each question is worth 10 points. The

criteria for evaluating each question on the test are as follows:

- 9-10 points – the answer is correct, complete, and well-reasoned;
- 7-8 points – the answer is correct, detailed, but not very well argued;

- 5-6 points – the answer is generally correct, but has shortcomings;
- 3-4 points – the answer contains minor errors;
- 1-2 points – the answer contains significant errors;
- 0 points – no answer or incorrect answer.

Maximum number of points for the module test: 10 points × 5 questions = 50 points.

The rating scale for the discipline is:

R = RC = 50 points + 50 points = 100 points.

Calendar control: conducted twice per semester as monitoring of the current status of syllabus requirements fulfillment.
At the first assessment (week 8), the student receives a "pass" if their current rating is at least 15 points (50% of the maximum number of points a student can receive before the first assessment).
At the second assessment (week 14), students receive a "pass" if their current rating is at least 20 points (50% of the maximum number of points a student can receive before the second assessment).
Semester control: credit
Conditions for admission to semester control:
With a semester rating of at least 60 points and all computer workshop assignments completed, the student automatically receives a credit according to the table (Table of correspondence of rating points to grades on the university scale). Otherwise, they must complete a credit test.
A prerequisite for admission to the credit test is the completion and defense of the computer workshop.
If a student does not agree with the automatic grade, they can try to improve their grade by writing a credit test, in which case their points earned during the semester are retained, and the better of the two grades received by the student is awarded (a "soft" grading system).

Table of correspondence between rating points and grades on the university scale

Number of points	**Grade**
100-95	Excellent
94	Very good
84	Good
74-65	Satisfactory
64-60	Sufficient
Less than 60	Unsatisfactory
Admission requirements not met	Not admitted

9. Additional information on the discipline (educational component)

Description of material, technical, and informational support for the discipline Computer lab of the Department of Radio Engineering Systems

---

Work program of the academic discipline (syllabus):
Compiled by Katin P. Yu.;
Approved by the Department of Radio Engineering Systems (Minutes No. 06/2024 dated 06/27/2024)
Approved by the methodological commission of the faculty/research institute (protocol No. 06/2024 dated 28.06.2024)