

Faculty of Computer Technology and Cybersecurity

Department of Computer Engineering

Approved

Dean of faculty

IITU JSC

\_\_\_\_\_\_\_\_\_\_\_\_\_N.A. Seilova

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2023

**SYLLABUS**

**(ACADEMIC PROGRAM)**

**Course (code, title):** SFT6306 Software Architecture and Design

**Major (code, title)**: 6B06106 Computer Systems and Software Engineering , 6B06110 Software Engineering

**Educational program** B057 Information technologies

**Year:** 1 **Semester:** 2 **Number of credits**: 4 ECTS

**Lectures:** 15 hours

# Laboratory classes: 30 hours

**T/SIS:** 75 hours

**Total:** 120 hours

**Cycle: (GER (university disciplines), BS, AS, electives)** university disciplines

# Final assessment form: Project

Almaty 2023

«IITU» JSC

Academic program of the course (code, title) SFT6306 Software Architecture and Design has been developed based on Standard Academic Program.

Academic program has been reviewed at the meeting of Computer Engineering department.

Minutes №. \_\_\_ dated «\_\_\_» \_\_\_\_\_\_2023

Head of the Department\_\_\_\_\_\_\_\_\_\_\_ PhD, ass.prof., Chinibayeva T.T.

Author \_\_\_\_\_\_\_\_\_\_\_\_ Assistant professor, Kayimov S.

\_\_\_\_\_\_\_\_\_\_\_\_ Senior lector Mamanova S.E.

The working academic program was approved at the meeting of the faculty academic quality council faculty of Computer Technology and Cybersecurity

Minutes № \_\_ dated " \_\_\_"\_ \_\_\_\_\_ 2023

Department \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ A.Ajibayeva

*Signature*

for Educational and Methodological Affairs

*F-75, Syllabus (Academic program)*

|  |  |
| --- | --- |
| **1. General information** | |
| Faculty | Computer Technology and Cybersecurity |
| Major code and title | SFT6306 Software Architecture and Design |
| Year, semester | 1,2 semester |
| Subject category | General education, basic, major |
| Number of Credits | 4 |
| Language of Delivery: | English |
| Prerequisites: | Application Development (SDP2), Performance, Data Structures & Algorithms (SDP4) |
| Postrequisites | Diploma Project |
| Lecturer | Name: Kayimov S. E-mail: s.kayimov@iitu.edu.kz  Mamanova S.E. E-mail: s.mamanova@iitu.edu.kz  Room 409, IITU, Almaty, Kazakhstan E-mail: |
| Instructors | 1) Kayimov S.  E-mail: s.kayimov@iitu.edu.kz  2) Mamanova S.E.  E-mail: s.mamanova@iitu.kz,  Room 409, IITU, Almaty, Kazakhstan |
| **2. Goals, objectives and learning outcomes of the course** | |
| **Course goals**  Formation of knowledge on the basics of software architecture, software development methodologies, different approaches for evaluate the risks during software development, testing of the software, UML design notations, architectural styles and design patterns.    **Course objectives:**   * To develop the students’ understandings of different software development methodologies and life cycle software development * To develop the students’ skills to use architectural styles and different design patterns. * To develop the students’ skills of properly creation of UML design notation * To foster the students’ interest in learning, develop their logical and critical thinking in software development     **Learning outcomes**  After completing this course, students will be able to demonstrate the following knowledge and skills, assessed through class discussions and assessments:   * to identify risks and discuss how to mitigate them in the software development * discuss architectural choices, the short-term and long-term consequences associated with each choice and the rationale for selecting one choice over the others. * to identify the flow of a system solution, from requirements to quality attributes and architectural structures to design patterns and detailed design, to implementation, testing, integration, sustainment, and future reengineering as required to extend a system’s life. * draft detailed design documents consistent with a specified architecture for moderate to small systems using UML design notations. * use standard OO and other requirement notions to understand stakeholder requirements and express those. * demonstrate understanding design intent required to implement modules, subsystems and systems. * create views to capture and communicate key aspects of a design element for a specific and targeted audience * use architectural styles and design patterns. | |
| **3. Learning outcomes of the CSSE educational program** | |
| LO1: Demonstrate solutions using basic mathematical tools to solve professional problems.  LO2: Analyze the structure of the main components of a computer, use a wide range of internal and external memory technologies; write program code to process bits in the processor.  LO3: Apply data structure and develop appropriate algorithms to solve various computing problems.  LO4: Apply various tools to develop software, user interface, storage and data processing systems.  LO5: use various software development methodologies, compile software documentation, apply required diagrams, develop models of the logical and physical structure of a software system, database, manage development progress.  RO6: Develop effective data storage systems and methods for their processing and analysis using machine learning algorithms.  LO7: Proficient in technologies for administering systems and networks of any configuration, troubleshooting and preventing threats.  LO8: Develop, operate and maintain robotic systems.  LO9: Demonstrate skills in developing complex 3D visualizations using computer-assisted lighting, augmented reality, and reality technologies.  LO10: Independently analyze modern sources in a comprehensive and rigorous manner, draw conclusions, give reasons for them and make decisions based on the information. | |
| **4. Learning outcomes of the SE educational program** | |
| LO1. Demonstrate the ability to use basic mathematical tools.  LO2. Apply various tools to develop software, user interfaces, and data storage and processing systems.  LO3. Explain the execution of programs in a high-level language at the instruction level; Use a wide range of memory technologies, internal and external; Write program code to manipulate bits in the processor.  LO4. Solve practical problems by creating programs in a good style, and modify and rewrite the created program using analysis tools, development environment(s) for creating and debugging applications, modern compiler environments.  LO5 Explain compiled software documentation and compose documentation using operation diagrams, class diagrams, state diagrams, entity relationship (ER) diagrams. Be able to develop models of the logical and physical architecture of a software system.  LO6. Design logical database schemas using relational, object-oriented, object-relational, key-value schemas for simple and complex defined systems.  LO7. Understand the software development life cycle, various software development methodologies and the place of testing in this process.  LO8. Ability to create test cases and generate test kits, develop and write acceptance tests, test scripts, and document defects found.  LO9. Have skills in selecting, designing, implementing, assessing the quality and analyzing the effectiveness of software to solve problems in various subject areas.  LO10 Independently analyze modern sources in a comprehensive and critical manner, draw conclusions, give reasons for them and make decisions based on the information. | |
| **5. Course description** | |
| This course lays a solid foundation upon which solutions for these much larger and more important applications may be built. Students will study large systems and how they were partitioned into subsystems and components, as well as how the structuring of these elements into a solution and the interfaces used to join them together facilitates communication and control. Students will explore with various notations and formalisms as they learn the relationship between these structures and key quality attributes and their impact on system implementation. Differences between detailed design and architecture are explored, as well as notations used for both. Two major applications are analyzed and the impact of several wellknown architectural styles is evaluated. The use of various notations is explored, with a focus on UML, and the role of architecture and detailed design specifications are considered from the perspective of risk management. | |
| **6. Course policy** | |
| **CLASS RULES**  Respect the learning environment:   * Come to class on time and prepared to discuss the assigned reading. * Do not distract the class with excessive private conversations. * Turn off all mobile devices. * Use the lab computers in the room for exercises only. * Bring paper and pens/pencils.   Attendance should be regular. The student gains points for each performed assignments. In case the student is not able to attend the lessons for some reason, he will be responsible fro learning all material, which was learnt during unattended lesson. If some student did not attend more than 20% of the lesson without reasonable excuse, the student will not be admitted to exam. Cheating will not be tolerated. Students caught cheating will receive “0” for assihnment.  **CLASS FORMAT AND SETUP**  In a typical week you will be introduced to a new concept, be assigned readings to be completed before class, will complete a group exercise that will require you to apply the tools and/or techniques related to the week’s concept, and will be given an individual assignment to complete as a project task.  **ONLINE PLATFORMS**  MS Teams, platonus.iitu.edu.kz | |
| **7. Literature Basic** literature:   1. Just Enough Software Architecture: A Risk-Driven Approach by George Fairbanks, Marshall & Brainerd Publishers, 2010. 2. Sommerville, L. , Software engineering, Ninth Edition / Lan Sommerville.- USA: Pearson, 2011. 3. DAMA-DMBOK: Data Management Body of Knowledge. Ackerman Anderson, Linda and Dean Anderson. 2017.     Supplementary literature:   1. Richard, H., Software Engineering, The Development Process / H. Richard, M.J.and Thayer Chiristensen; Edited by: Richard H. and Mark J. Chiristensen. Foreword by: Carl K. Chang.- Third edition.- Wiley-Interscience, 2005. 2. Len Bass, Paul Charls Clements, Rick Kazman. Software Architecture in Practice. 3rd Edition.-Addison-Wesley Professiobal, 2012.-620 p. 3. Design Patterns. Elizabeth Robson, Eric Freeman. O'Reilly Media, Inc. October, 2004.   Recommended MOOC on Cousera:  1. Software Design and Architecture (https://www.coursera.org/specializations/softwaredesign-architecture) | |

**Course Content**

# 1. Lecture, practical/seminar/laboratory session plans

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| TSIS | Teacher-supervised independent study (СРСП) |
| SIS | Students’ independent study (СРС) |
| P | Project |
| PA | Practical assignment |
| LW MCQ | Laboratory Work  Multiple choice quiz |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Week No** | **Lectures (1 h/w) Course Topic** | **Reference**  **Materials** | Lectures(1 h/w) | Lab. Sessions (1 h/w) | TSIS(1 h/w) | SIS (1 h/w) |
| 1 | Introduction. Requirements  Engineering. Basics of Use Cases | Basic literature:   * Chapter 1 of Fairbanks * Chapter 9.5.1 (pp. 143-145) of   Fairbanks  Supplementary literature:   * Chapter 1 of Sommerville • Chapter 5 (pp.   144-146) of  Sommerville | 1 | LW 1 | Project 1 | Assig ned  readin gs |
| 2 | Risk Assessment and Mitigation. | Basic literature:   * Chapters 3-5 of   Fairbanks  Supplementary literature:   * Chapter 4 of   Sommerville | 1 | LW 2 | Project 1 | Projec  t 1 |
| **Week No** | **Lectures (1 h/w) Course Topic** | **Reference**  **Materials** | Lectures(1 h/w) | Lab. Sessions (1 h/w) | TSIS(1 h/w) | SIS (1 h/w) |
| 3 | Software Development Life  Cycle (SDLC) | Supplementary literature:  • Chapter 2-3 of Sommerville | 1 | LW 3 | Project 1 | Projec  t 1 |
| 4 | Architecture Basics | Basic literature:   * Chapter 2, 6 of   Fairbanks    Supplementary literature:   * Chapter 6 of Sommerville   (pp.168-173) | 1 | LW 4 | Project 1 | Projec t 1 |
| 5 | Software Abstractions | Basic literature:  • Chapter 6, 7 of  Fairbanks, Chapter  13 is optional    Supplementary literature:  • Chapter 5 of  Sommerville | 1 | LW 5 | Project 2 | Projec t 2 |
| 6 | Architectural Modeling. Domain Model. Design Model | Basic literature:   * Chapter 7, 8 of   Fairbanks   * Chapter 9 of   Fairbanks, Chapter  10 is optional    Supplementary literature:  • Chapter 5 of  Sommerville | 1 | LW 6 | Project 2 | Projec  t 2 |
| 7 | Analyses existing architectural solutions(best practice) | Basic literature:   * Chapter 8 of   Fairbanks   * Chapter 9 of   Fairbanks, Chapter  9 is optional | 1 | Midterm | Project 2 | Projec t 2 |
| **Week No** | **Lectures (1 h/w) Course Topic** | **Reference**  **Materials** | Lectures(1 h/w) | Lab. Sessions (1 h/w) | TSIS(1 h/w) | SIS (1 h/w) |
| 8 | Creating the Model.UML  Basics | Supplementary literature:   * Chapter 5, 7.1 of Sommerville * IBM Developer Works Article on   Introduction to  UML   * https://develope   r.ibm.com/technol ogies/webdevelopment/articl es/an-introductionto-uml/ | 1 | LW 7 | Project 3 | Projec t 3 |
| 9 | UML Scenarios and Sequence Diagrams | Supplementary literature:   * Chapter 5, 7.1 of Sommerville * IBM Developer Works Article on   Introduction to  UML  https://developer.ib m.com/technologie s/web-  development/article s/an-introductionto-uml/ | 1 | LW 8 | Project 3 | Projec t 3 |
| 10 | UML Component Diagrams | Supplementary literature:  • Chapter 5, 7.1 of Sommerville • IBM Developer Works Article on  Introduction to  UML  https://developer.ib m.com/technologie s/web-  development/article s/an-introductionto-uml/ | 1 | LW 9 | Project 3 | Projec t 3 |
| 11 | UML Class Diagrams | Supplementary literature:   * Chapter 5, 7.1 of Sommerville * IBM Developer Works Article on   Introduction to  UML  https://developer.ib m.com/technologie s/web-  development/article s/an-introductionto-uml/ | 1 | LW  10 | Project 3 | Projec t 3 |
| **Week No** | **Lectures (1 h/w) Course Topic** | **Reference**  **Materials** | Lectures(1 h/w) | Lab. Sessions (1 h/w) | TSIS(1 h/w) | SIS (1 h/w) |
| 12 | Design Patterns | Basic literature:   * Chapter 14 of   Fairbanks    Supplementary literature:   * Chapter 6.3 of Sommerville * http://www.ood esign.com/designprinciples.html * http://www.dev eloper.com/design/ article.php/147456 1/What-AreDesign-Patternsand-Do-I-NeedThem.htm | 1 | LW  11 | Project 4 | Projec t 4 |
| 13 | Architectural Styles | Basic literature:   * Chapter 14 of   Fairbanks    Supplementary literature:   * Chapter 6.3, 7.2 of Sommerville * http://www.ood esign.com/designprinciples.html * http://www.dev eloper.com/design/ article.php/147456 1/What-AreDesign-Patternsand-Do-I-NeedThem.htm | 1 | LW  12 | Project 4 | Projec t 4 |
| 14 | Characteristics of Good Models | Basic literature:  • Chapter 15 of  Fairbanks | 1 | End-  Term | Project 5 | Projec t 5 |
| 15 | Bonus Topic | Basic literature:  • Chapter 16 of  Fairbanks | 1 | Defen  ce of Projec  t | Project 5 | Projec t 5 |
|  | **Total hours** | **120** | **15** | **30** | **15** | **60** |

# 2. List of topics/ assignments for laboratory classes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Topic Title** | **Number of hours** | **References** | **Form of reporting** | **Deadline** |
| **1** | **2** | **3** | **4** | **5** | **6** |
| 1 | Software requirments specification. Identifying Use Cases | 2 | BL, SL | On-line | week 1 |
| 2 | Prioritizing and Mitigating Risks | 2 | BL, SL | On-line | week 2 |
| 3 | Software life cycle modeling | 4 | BL, SL | On-line | week 3 |
| 4 | Software Abstractions. Analyze Architecture Diagrams. | 2 | BL, SL | On-line | week 5 |
| 5 | Architectural Modeling | 4 | BL, SL | On-line | week 6 |
| 6 | Dynamic UML diagrams | 2 | BL, SL | On-line | week 8 |
| 7 | UML Class, Component Diagrams | 2 | BL, SL | On-line | week 9 |
| 8 | UML Deployment Diagram | 4 | BL, SL | On-line | week 10 |
| 9 | Design Patterns | 4 | BL, SL | On-line | week 12 |
| 10 | Architectural Styles | 4 | BL, SL | On-line | week 13 |
|  | **Total** | **30** |  |  |  |

**3. Сriteria for evaluating laboratory work adopted by the academic credit system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Letter**  **Grade** | **Numerical equivalent** | **Points (%)** | **Grading scheme** | **General description of evaluation criteria** |
| A | 4,0 | 95-100 | Excellent | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; All laboratory tasks were completed without errors. The student completed everything according to the standard and submitted it on time. Answered all questions and described all the work done in detail. |
| A- | 3,67 | 90-94 | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; gives an exhaustive answer. |
| B+ | 3,33 | 85-89 | Good | The student completed all the tasks from the laboratory work completely correctly. The student shows complete, sufficiently substantiated knowledge of the laboratory work, however, the answers did not always highlight the main thing. The answers were mostly concise and not always clear. The student was unable to describe all the work done and methods for solving problems. |
| B | 3,0 | 80-84 |
| B- | 2,67 | 75-79 |
| C+ | 2,33 | 70-74 |
| C | 2,0 | 65-69 | Satisfactory | The student completed all the tasks from the laboratory work completely correctly, formatted his work according to the standard, and added his work and defended it on time. But theoretical knowledge is not sufficiently mastered, the student could not describe the work done |
| C- | 1,67 | 60-64 |
| D+ | 1,33 | 55-59 |
| D | 1,0 | 50-54 |
| F | 0 | 0-49 | FX (25-49) Fail with re-exam | The student demonstrates insufficient knowledge of the laboratory work, The laboratory work was not completed according to the standard and the tasks were not completed on time. |
| F (0-24) Fail | The student demonstrates a very low level of knowledge of the laboratory work. |

# 4. List of topics/assignments for Student Independent Study

Proper organization of students independent study is the key to the formation of skills in mastering, learning, assimilation and systematization of acquired knowledge, ensuring a high level of academic performance in the learning process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Assignments (topics) for Independent study | Hours | Recommended literature and other sources (links) | Form of submission |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Software requirements specification. | 7 | BL, SL | On-line, Oral defense |
| 2 | Risk-Driven Model | 7 | BL, SL | On-line, Oral defense |
| 3 | Software Development Life Cycle | 7 | BL, SL | On-line, Oral defense |
| 4 | Software Architecture Models | 7 | BL, SL | On-line, Oral defense |
| 5 | Software Abstractions | 7 | BL, SL | On-line, Oral defense |
| 6 | Canonical Model | 7 | BL, SL | On-line, Oral defense |
| 7 | Encapsulation and Partitioning | 7 | BL, SL | On-line, Oral defense |
| 8 | Creating the Model. UML Diagrams | 7 | BL, SL | On-line, Oral defense |
| 9 | Design Principles, Design Patterns | 7 | BL, SL | On-line, Oral defense |
| 10 | Architectural Styles | 7 | BL, SL | On-line, Oral defense |
| 11 | Refining and Using Architectural Models | 5 | BL, SL | On-line, Oral defense |
| **Total** | | **75** |  |  |

**5. Сriteria for evaluating student independent study adopted by the academic credit system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Letter**  **Grade** | **Numerical equivalent** | **Points (%)** | **Grading scheme** | **General description of evaluation criteria** |
| A | 4,0 | 95-100 | Excellent | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; All tasks were completed without errors. The student completed everything according to the standard and submitted work on time. Answered all questions and described all the work done in detail. |
| A- | 3,67 | 90-94 | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; gives an exhaustive answer. |
| B+ | 3,33 | 85-89 | Good | The student completed all the tasks from the SIS completely correctly. The student shows complete, sufficiently substantiated knowledge of the work, however, the answers did not always highlight the main thing. The answers were mostly concise and not always clear. The student was unable to describe all the work done and methods for solving problems. |
| B | 3,0 | 80-84 |
| B- | 2,67 | 75-79 |
| C+ | 2,33 | 70-74 |
| C | 2,0 | 65-69 | Satisfactory | The student completed all the tasks from the student independent study work completely correctly, formatted his work according to the standard, and added his work and defended it on time. But theoretical knowledge is not sufficiently mastered, the student could not describe the work done |
| C- | 1,67 | 60-64 |
| D+ | 1,33 | 55-59 |
| D | 1,0 | 50-54 |
| F | 0 | 0-49 | FX (25-49) Fail with re-exam | The student demonstrates insufficient knowledge of the student independent study work, The work was not completed according to the standard and the tasks were not completed on time. |
| F (0-24) Fail | The student demonstrates a very low level of knowledge of the student independent study work. |

# 6. Student performance evaluation system for the course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Period** | **Assignments** | **Maximum score** | **Coefficient** | **Total** |
| 1st attestation | **laboratory works:**  LW 1  LW 2  LW 3  LW 4  LW 5  **Mid term** | 100  100  100  100  100  100  100 | 0,75  0,15  0,15  0,15  0,15  0,15  0,25 | 100 |
| 2nd attestation | **laboratory works:**  LW 6  LW 7  LW 8  LW 9  LW10  **End of term** | 100  100  100  100  100  100  100 | 0,75  0,15  0,15  0,15  0,15  0,15  0,25 | 100 |
| Final exam | Project |  |  | 100 |
| **Total** | **0,3\*1stAtt+0,3\*2ndAtt+0,4\*Final** | | | **100** |

\*If the number of absences exceeds 20%, student will be automatically scheduled for a Retake

(summer semester)

**7. Achievement level as per course curriculum shall be assessed according to the evaluation chart adopted by the academic credit system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Letter**  **Grade** | **Numerical equivalent** | **Points (%)** | **Grading scheme** | **General description of evaluation criteria** |
| A | 4,0 | 95-100 | Excellent | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; shows a high level of knowledge in excess of the amount provided by the syllabus, gives an exhaustive answer. |
| A- | 3,67 | 90-94 | The student owns the knowledge of the subject in full the curriculum, deeply comprehends the discipline; gives an exhaustive answer. |
| B+ | 3,33 | 85-89 | Good | The student shows complete, sufficiently substantiated knowledge of the subject, however, the answers did not always highlight the main thing, rational calculation methods were not always used; the answers were mostly concise and not always clear. |
| B | 3,0 | 80-84 |
| B- | 2,67 | 75-79 |
| C+ | 2,33 | 70-74 |
| C | 2,0 | 65-69 | Satisfactory | The student demonstrates sufficient knowledge of the subject, but without the proper depth and justification, the answers are fuzzy and without the proper logical sequence. |
| C- | 1,67 | 60-64 |
| D+ | 1,33 | 55-59 |
| D | 1,0 | 50-54 |
| F | 0 | 0-49 | FX (25-49) Fail with re-exam | The student demonstrates insufficient knowledge of the subject, on the individual issues are not given a positive response. |
| F (0-24) Fail | The student demonstrates a very low level of knowledge of the subject. |