



ІНФОРМАЦІЙНІ СИСТЕМИ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>11 Математика і статистика</i>
Спеціальність	<i>113 Прикладна математика</i>
Освітня програма	<i>Наука про дані та математичне моделювання</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>4 курс, осінній семестр</i>
Обсяг дисципліни	<i>5</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен/МКР (1) та лабораторні роботи (4)</i>
Розклад занять	<i>Лекції – 1 раз на тиждень (18 лекцій) Лабораторні заняття – 1 раз на тиждень (18 занять)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: Шияк Богдан Антонович Лабораторні: Ковальчук-Химюк Людмила Олександрівна, Шияк Богдан Антонович</i>
Розміщення курсу	<i>Слак-канали #dbis-course та #dbis-labs</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни "Інформаційні системи" дає змогу сформувати у студентів професійні компетенції, необхідні для розв'язання задач, пов'язаних із побудовою інформаційних систем, починаючи з аналізу вимог і завершуючи розгортанням системи для кінцевих користувачів.

Метою дисципліни є опанування студентом основних принципів аналізу функціональних та нефункціональних вимог, побудови архітектури інформаційної системи на основі зібраних вимог та реалізації запропонованої архітектури через створення інформаційної систем.

Предметом дисципліни є:

- підходи до роботи з функціональними та нефункціональними вимогами;
- підходи до побудови та опису архітектури системи;
- архітектурні стилі та підходи, що дозволяють більш ефективно досягати тих чи інших якісних атрибутів системи;
- сучасні технологічні підходи до реалізації інформаційних систем.

Результатами освоєння дисципліни є:

- вміння аналізувати проблему та ідентифікувати бізнес вимоги та необхідності;
- вміння на основі ідентифікованих бізнес вимог формувати вимоги до інформаційної системи;
- вміння обирати та обґрунтовувати архітектурне бачення системи;
- знання сучасних моделей розповсюдження програмних продуктів;

- вміння обирати оптимальне рішення для розгортання системи.
- 2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Проходження курсу вимагає від студента базового володіння хоча б однією мовою програмування (наприклад, Python, Java, Javascript) та вміння працювати з базами даних (наприклад, PostgreSQL чи MySQL). Таким чином дана дисципліна використовує знання на уміння отримані під час вивчення кредитних модулів "Програмування на мові Python", "Вступ до баз даних та інформаційних систем" та "Аналіз даних".

3. Зміст навчальної дисципліни

Розділ 1. Робота з вимогами

1. Основи бізнес архітектури
2. Робота з зацікавленими особами (stakeholder management)
3. Функціональні вимоги та способи роботи з ними
4. Нефункціональні вимоги та способи роботи з ними

Розділ 2. Організацію процесу розробки інформаційних систем

1. Гнучкі методології та їх вплив на архітектурні рішення
2. Організація процесу командної розробки
3. Формування MVP

Розділ 3. Управління архітектурою системи

1. Ведення документації системи в умовах гнучкої розробки
2. Підготовка представлень архітектури системи для різних класів зацікавлених осіб
3. Формування архітектури системи на основі підходу C4
4. Формування архітектури системи на основі підходу 4+1
5. Основні архітектурні стилі
6. Побудова REST API

Розділ 4. Розгортання системи

1. Організація кодової бази інформаційної системи
2. Принципи неперервної інтеграції та доставки змін (Continuous Integration/Continuous delivery)
3. Віртуалізація та контейнеризація розподілених систем
4. Принципи побудови cloud-native систем
5. Шаблони проектування мікросервісних систем

4. Навчальні матеріали та ресурси

Базові джерела:

- Матеріали курсу, що публікуються викладачами у слак каналі курсу
- Лавріщева К.М. Програмна інженерія — Київ — 2008 р. — 319 с. (<http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf>)
- Принципи The Twelve-Factor App (<https://12factor.net/uk/>)

Додаткові джерела:

- A Conceptual Model of Architecture Description (<http://www.iso-architecture.org/42010/cm/index.html>)
- C4 model (<https://c4model.com/>)
- Architectural styles and the design of network-based software architectures RT Fielding - PhD Thesis, University of California, 2000 (https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)

- *Architectural Blueprints—The “4+1” View Model of Software Architecture*, by Philippe Kruchten (<https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>)
- <https://www.viewpoints-and-perspectives.info/home/viewpoints/>
- *Software Architecture in Practice (2nd Edition)*, by Len Bass, Paul Clements, Rick Kazman, Addison-Wesley, 2003 (<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30659> або <https://people.ece.ubc.ca/~matei/EECE417/BASS/>)
- *Patterns of enterprise application architecture*, by Martin Fowler, with Dave Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and Randy Stafford, Addison-Wesley, 2003.
- *Characterizing Architecturally Significant Requirements*, by Lianping Chen, Muhammad Ali Babar, Bashar Nuseibeh, 2013 (https://www.researchgate.net/publication/255569055_Characterizing_Architecturally_Significant_Requirements)
- *Architectural Decision Records* (<https://adr.github.io/>)
- *Шаблони мікросервісних архітектур* (<https://microservices.io/patterns/index.html>)
- *Software Architecture Patterns*, by Mark Richards, O'Reilly Media, Inc., 2015 (<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/>)
- *BoundedContext*, by Martin Fowler (<https://martinfowler.com/bliki/BoundedContext.html>)
- *Microservices*, by James Lewis, Martin Fowler (<https://martinfowler.com/articles/microservices.html>)
- *A successful Git branching model*, by Vincent Driessen (<https://nvie.com/posts/a-successful-git-branching-model/>)
- *Richardson Maturity Model*, by Martin Fowler (<https://martinfowler.com/articles/richardsonMaturityModel.html>)

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекції

Розділ 1. Робота з вимогами

Лекція 1. Бізнес Архітектура. Бізнес драйвери, мета, цілі. Робота з зацікавленими особами (stakeholder management), PI-матриця, RACI-матриця.

Лекція 2. Робота з вимогами. Класифікація вимог. Способи управління функціональними вимогами.

Лекція 3. Нефункціональні вимоги. Класифікація. Способи запису. Архітектурно значущі вимоги.

Розділ 2. Організацію процесу розробки інформаційних систем

Лекція 4. Методології розробки програмного забезпечення. XP. Scrum.Kanban

Лекція 5. Мінімально життєздатний продукт (MVP). Структурна декомпозиція робіт (WBS). Оцінка об'єму проекту.

Розділ 3. Управління архітектурою системи

Лекція 6. Документування архітектури інформаційної системи. Модель 4+1. Модель C4. ADR.

Лекція 7. Архітектурні стилі. Моноліт. SOA. Мікросервіси. Слої. Архітектура побудована на подіях.

Лекція 8. Принципи побудови REST API.

Лекція 9. REST, як архітектурний стиль.

Розділ 4. Розгортання системи

Лекція 10. Організація кодової бази інформаційної системи. Принципи неперервної інтеграції та доставки змін.

Лекція 11. Принципи побудови cloud-native систем. 12 factor apps.

Лекція 12. Віртуалізація та контейнеризація розподілених систем. Docker. Оркестрація контейнерів.

Лекція 13. Мікросервіси, як архітектурний стиль. Переваги мікросервісних архітектур. Типові ризики та недоліки мікросервісних архітектур.

Лекція 14. Шаблони проектування мікросервісних систем. Синхронна та асинхронна комунікація. Надійність. Безпека та аудит.

Лабораторні роботи

Передбачено 4 лабораторних роботи, по одній для кожного розділу.

Розділ 1.

1. Описати

- a. Business drivers (можуть бути вигадані командою, але не можуть змінюватись після їх оприлюднення на момент здачі роботи)
- b. Goals
- c. Objectives

2. Побудувати граф зв'язків між Business drivers, Goals, Objectives.

3. На основі бізнес-цілей та документів опублікованих у слак:

- a. Побудувати RACI матрицю для курсового проекту.
- b. Описати яку зацікавленість (Concerns) має кожен stakeholder у проекті та які представлення (view) необхідно для нього приготувати.
- c. Побудувати сценарії використання для основних бізнес процесів.
- d. Описати історії користувача, використовуючи будь-який шаблон
- e. Описати нефункціональні вимоги у вигляді quality attribute scenarios. Має бути покрито щонайменше 3 якісних атрибути (quality attributes).

Розділ 2.

1. Ідентифікувати усі функціональні вимоги до системи на основі власних очікувань.
2. Ідентифікувати усі нефункціональні вимоги до системи на основі власних очікувань від системи та документів за посиланням у слак каналі. До кожного NFR вказати до якої групи він відноситься
3. Визначити ASR.
4. Сформулювати список функціональності, що буде включено до Мінімально життєздатного продукту (MVP).
5. Підготувати структуру декомпозиції робіт (WBS) для MVP.
6. Підготувати структуру декомпозиції робіт (WBS) для всього проекту.

Опис кожної активності в WBS має містити відповідального за реалізацію члена команди.

Розділ 3.

1. Розгортання MVP на Негоки та демонстрація. Рекомендації до формату демонстрації:
 - a. 15 хв на команду (важливо, щоб команда вклалась у цей час);
 - b. лідер робить презентацію проекту, націлену на "бізнес людей" та "високорівневий IT-менеджмент" потенційного замовника;

- c. кожний із студентів розповідає про зроблену ним роботу (на основі цього буде виставлятися оцінка за лабораторну)
2. *Задокументована архітектура системи, що включає в себе:*
 - a. *Моделі даних;*
 - b. *Структура компонентів системи (модулів, мікросервісів і тд.);*
 - c. *Інтерфейси взаємодії компонентів між собою, та інтерфейси взаємодії з системою;*
 - d. *Схему розгортання системи.*

Для презентації 1.b можна використовувати схеми підготовлені для пункту 2. Уся документація має бути у форматі pdf.

Для підготовки діаграм можна використовувати будь-який зручний для студентів редактор, але до перевірки діаграми мають бути надані у форматі pdf. Приклади сервісів:

- <https://www.lucidchart.com/pages/usecase/education>
- <https://app.diagrams.net/> (draw.io)

Розділ 4.

1. *Створити всі необхідні Docker образи (як для сервісів з бізнес логікою, так і для серверів БД). Для серверів БД можна використовувати стандартні образи з <https://hub.docker.com/search?q=&type=image&category=database>*
2. *Розгорнути MVP у Docker compose. На вибір студентів можна демонструвати як на девелоперських машинах, так і розгорнути на будь-якому хостингу.*
3. *Дані в БД мають бути збережені при перезапуску Docker compose.*

Під час демонстрації роботи студенти мають продемонструвати відображення логів сервісів. По логах має бути зрозуміло через які компоненти системи пройшов запит користувача.

6. Самостійна робота студента/аспіранта

До самостійного опрацювання виносяться:

- *Architectural Blueprints—The “4+1” View Model of Software Architecture, by Philippe Kruchten*
- *Architectural styles and the design of network-based software architectures RT Fielding - PhD Thesis, University of California, 2000*

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- **Відвідування лекцій та лабораторних занять.** Відсутність на лекціях та лабораторних заняттях без поважної причини не допускається.
- **Правила поведінки на заняттях.** На лекційних та лабораторних заняттях студенти мають вимкнути мобільні телефони або увімкнути їх на беззвучний режим. Забороняється користуватися телефонами та іншими засобами зв'язку для пошуку інформації на гугл-диску викладача, в Інтернеті тощо.
- **Правила захисту лабораторних робіт.** Усі лабораторні роботи оформляються і здаються студентами у вигляді звітів – doc або pdf-файлів. Захист лабораторних робіт відбувається у вигляді усної співбесіди зі студентом за результатами оформленого звіту, також студенту можуть бути поставлені запитання зі списку контрольних питань.

- **Правила призначення заохочувальних та штрафних балів.** За проходження онлайн курсів, погоджених з викладачем, можуть бути нараховані бонусні бали, але не більше ніж 10% від максимального балу за семестр (відповідно до РСО). Штрафні бали призначаються за несвоєчасне виконання лабораторних робіт (відповідно до РСО)
- **Політика щодо академічної доброчесності.** Згідно з Кодексом честі студента КПІ, при виконанні лабораторних чи контрольних робіт **забороняється** користуватися чужими виконаними лабораторними чи контрольними роботами та їх фрагментами. У випадку виявлення плагіату лабораторна робота може бути оцінена від 0 до 1/3 сумарної кількості запланованих за неї балів. Під час виконання контрольних робіт забороняється користуватись будь-якими сторонніми джерелами.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Вказуються всі види контролю та бали за кожен елемент контролю, наприклад:

Поточний контроль: МКР, лабораторні роботи

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: *екзамен*

Умови допуску до семестрового контролю: *зарахування усіх лабораторних робіт та семестровий рейтинг рівний або більше 30 балів.*

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- *Курсом передбачена можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за відповідною тематикою.*

Робочу програму навчальної дисципліни (силабус):

Складено асистент Шияк Б.А.

Ухвалено кафедрою *прикладної математики* (протокол № 13 від 16.06.2021)

Погоджено Методичною комісією факультету *прикладної математики* (протокол № 9 від 24.06.2021)