МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до виконання курсового проєкту для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня

Укладачі: **Золотарьова І.О.**

Беседовський О.М.

Відповідальний за видання **Бондаренко Д.О.**

Харків
ХНЕУ ім. С. Кузнеця
2024

УДК 004.415 (07)
ББК 32.973-018р

Укладачі: Золотарьова І.О.,
Беседовський О.М.

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 7 від 21.12.2023 р.

Самостійне електронне текстове мережеве видання

075 Інженерія програмного забезпечення: методичні рекомендації до виконання курсового проєкту для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня / укл. Золотарьова І.О., Беседовський О.М. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2024. - 36 с. (Укр. мов.)

Викладено питання організації курсового проєктування, наведено вимоги до структури курсового проєкту та оформлення пояснювальної записки, методичні рекомендації до розроблення його структурних елементів.

Рекомендовано для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення".

© Харківський національний економічний
університет імені Семена Кузнеця, 2024

Вступ

В сучасному світі програмне забезпечення стає не лише ключовою складовою технологічного прогресу, але й необхідним елементом в різних сферах життя, починаючи від промисловості та бізнесу і закінчуючи особистими пристроями. Отже, важливість розуміння та володіння методами та інструментами інженерії програмного забезпечення стає безсумнівною. З цією метою розроблено методичні рекомендації, спрямовані на підтримку студентів при виконанні цього курсового проєкту.

Виконання курсового проєкту з інженерії програмного забезпечення передбачає глибоке розуміння процесу розробки ПЗ, включаючи аналіз вимог, проєктування, моделювання тощо. Ці рекомендації мають на меті сприяти студентам в успішному виконанні курсового проєкту.

Важливим аспектом є системність підходу до розробки програмного забезпечення. Це означає не лише розуміння окремих кроків у процесі розробки, але й їх взаємозв'язок та вплив на якість кінцевого продукту. Методичні рекомендації допоможуть студентам у систематизації своїх знань та навичок, а також у здійсненні комплексного аналізу та управління різними етапами розробки програмного забезпечення.

Необхідним елементом для успішного виконання курсового проєкту з інженерії програмного забезпечення є самостійна робота здобувачів із технічною літературою, сучасними мовами, технологіями та середовищами, які спрямовані на покращення роботи фахівців на всіх етапах створення програмного продукту.

Курсовий проєкт є важливим етапом підготовки здобувачів. Він спрямований на більш глибоке осмислення і закріплення теоретичних знань, отриманих при вивченні вказаної дисципліни, та на вдосконалення практичних навичок щодо розроблення проєктної документації до програмного забезпечення, що має бути створено.

Ці методичні рекомендації встановлюють єдині правила та порядок виконання, оформлення і захисту курсового проєкту. Вони призначені для допомоги здобувачам спеціальності 121 "Інженерія програмного забезпечення" у виконанні курсового проєкту.

Результати навчання та компетентності, які формує та закріплює курсовий проєкт визначено в табл. 1.

Таблиця 1

Результати навчання та компетентності, які формує освітня компонента

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
РН 06	ЗК 01, ЗК 02, СК 10, СК 11
РН 14	ЗК 02, ЗК 03, СК 03, СК 04, СК 13

де, РН 06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

РН 14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ЗК 01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 02. Здатність застосовувати знання у практичних ситуаціях.

ЗК 03. Здатність спілкуватися державною мовою як усно, так і письмово.

СК 03. Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК 04. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

СК 10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК 11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК 13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

1. Мета й завдання курсового проєктування

Мета курсового проєкту з інженерії програмного забезпечення полягає у наданні здобувачам можливості засвоєння та практичного застосування ключових принципів, методів та інструментів розробки програмного забезпечення. Крім того, проєкт спрямований на розвиток навичок аналізу вимог, проектування архітектури програмного продукту.

Курсовий проєкт також ставить за мету навчити студентів ефективно працювати у команді, використовуючи сучасні методи колективної роботи та системи контролю версій. В результаті виконання проєкту здобувачі будуть здатні розробляти вимоги до програмного

забезпечення, яке відповідає вимогам замовника, має стабільну архітектуру.

Під час розроблення курсового проєкту здобувач має показати знання:

предметної області відповідно до постановки завдання;

процесів розробки програмного забезпечення;

методик розробки програмного забезпечення;

з принципів версійного контролю та колективної роботи.

Здобувач має уміти:

виконувати постановку завдання та її аналіз;

аналізувати та формулювати вимоги;

проектувати архітектуру системи;

працювати в команді;

управляти проєктом;

комунікації в процесі розробки;

розробляти та оформлювати необхідну документацію.

Робота над курсовим проєктом певною мірою визначає загальнотеоретичну та спеціальну підготовку здобувача і в остаточному підсумку готує його до майбутнього виконання більш складного й завершального етапу навчального процесу – дипломного проєктування. Здобувач має розглядати роботу над курсовим проєктом як своєрідну «репетицію» дипломного проєктування.

2. Організація курсового проєктування

Якісне виконання курсового проєкту вимагає чіткої організації роботи здобувача з моменту вибору теми проєкту й до його захисту.

Керівництво курсовим проєктуванням здійснюється викладачами кафедри інформаційних систем. Керівник курсового проєкту рекомендує здобувачу основну літературу, орієнтує його на розроблення необхідних проєктних рішень.

Здобувачу може бути призначена тема курсового проєкту з переліку рекомендованих тем (див. додаток А). Також йому надається право самостійного вибору теми проєкту. Якщо тема пропонується здобувачем, то вона має бути обговорена й погоджена з керівником курсового проєкту.

Матеріали виконаного курсового проєкту: пояснювальну записку в друкованому та електронному вигляді, презентацію доповіді в електронному вигляді здобувач має здати на перевірку керівникові не пізніше визначеного строку.

Захист курсових проєктів організовується кафедрою інформаційних систем за затвердженим графіком.

Під час захисту здобувач коротко доповідає про поставлене йому завдання, проєктні рішення, що були прийняти, одержані результати, відповідає на запитання. Доповідь має супроводжуватися демонстрацією презентації, яка має містити слайди, які визначає здобувач для більш повного розкриття теми.

3. Структура та обсяг курсового проєкту

Курсовий проєкт складається з пояснювальної записки та інших обов'язкових матеріалів (схеми, діаграми, графіки, таблиці, рисунки тощо), що розробляються відповідно до завдання.

Обсяг пояснювальної записки – 20 – 30 друкованих сторінок формату А4 (без додатків).

У табл. 1 наведено структура пояснювальної записки курсового проєкту.

4. Методичні рекомендації до розроблення структурних елементів пояснювальної записки курсового проєкту

Загальними вимогами до тексту пояснювальної записки є логічна послідовність викладення матеріалу, чіткість і конкретність викладення теоретичних і практичних результатів роботи, суті постановки завдання та мети роботи, методів дослідження, прийнятих рішень, доведеність висновків і обґрунтованість рекомендацій. У тексті пояснювальної записки необхідно дотримуватись єдиної термінології. Вона не має бути перевантажена малоінформативним матеріалом, описом загальновідомих даних, виведенням формул тощо. Необхідно посилатися на джерела інформації.

Таблиця 1

Структура пояснювальної записки курсового проєкту

Структурні елементи пояснювальної записки	Кількість сторінок
Титульний аркуш	1
Реферат	1
Зміст	1
Вступ	1
1 Бізнес-аналіз предметної області <назва предметної області>	4 - 10
1.1 Бізнес-вимоги (Business Requirements)	
1.1.1 Обґрунтування (Background)	
1.1.2 Можливість для бізнесу (Business Opportunity)	
1.1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)	
1.1.4 Потреби клієнтів або ринку (Customer or Market Needs)	
1.1.5 Бізнес-ризики (Business Risks)	
1.2 Бачення рішення (Vision of the Solution)	
1.2.1 Заява про бачення (Vision Statement)	
1.2.2 Основні характеристики (Major Features)	
1.2.3 Припущення та залежності (Assumptions and Dependencies)	
1.3 Сфера застосування та обмеження (Scope and Limitations)	
1.3.1 Обсяг початкового випуску застосунку (Scope of Initial Release)	
1.3.2 Обмеження та виключення (Limitations and Exclusions)	
1.4 Бізнес контекст (Business Context)	
1.4.1 Профілі зацікавлених сторін (Stakeholder Profiles)	
1.4.2 Операційне середовище (Operating Environment)	
2 Специфікація вимог до модуля (системи)	10 - 17
2.1 Аналіз існуючих рішень (аналогів, конкурентів)	
2.2 Специфікація функціональних вимог	
2.3 Нефункціональні вимоги	
2.4 Проектування інтерфейсу користувача (User interfaces)	
2.5 Деталізація нефункціональних вимог. Логічна структура даних (Detailed Non-Functional Requirements. Logical Structure of the Data)	
Висновки	1
Список використаних джерел	1
Додатки	

Текст пояснювальної записки не слід викладати від першої особи, переважно безособова форма (наприклад, «обчислюється», «знаходимо») за всім текстом у визначеному відмінку й часі.

При викладенні матеріалу не слід використовувати:

розмовні звороти;

жаргонні слова та звороти;

різні терміни для позначення одного поняття;

іншомовні слова та терміни за наявності в українській мові рівнозначних слів і термінів;

скорочення слів і словосполучень, крім встановлених правилами орфографії та нормативними документами.

Титульний аркуш є першою сторінкою пояснювальної записки. Він містить дані, які подають у такій послідовності:

відомості про виконавця роботи;

повна назва документа;

підписи відповідальних осіб, включаючи керівника роботи;

рік складення пояснювальної записки;

Приклад титульного аркушу наведено у додатку Б.

До змісту включають: вступ; послідовно перелічені назви всіх розділів, підрозділів та пунктів основної частини пояснювальної записки; висновки; перелік посилань; назви додатків і номери сторінок, які містять початок матеріалу.

Приклад змісту пояснювальної записки наведено у додатку В.

У вступі необхідно надати ідентифікувати та сформулювати проблему бізнесу, яка виникла в компанії, обґрунтувати актуальність теми проекту для вирішення цієї проблеми на основі розроблюваного модуля або системи. Коротко охарактеризувати функціональність модуля або системи. Необхідно охарактеризувати технічну та програмну платформу розробки автоматизованого модуля. Потрібно сформулювати мету та задачі проекту, визначити об'єкт і описати предметну область проектування. Також необхідно навести інформацію щодо засобів проектування, які використовувались у курсовому проєкті, та можливих галузей застосування результатів, отриманих у курсовому проєкті.

Перший розділ «Бізнес-аналіз предметної області <назва предметної області>». Метою розділу є проведення детального аналізу

проблеми, яка виникла на об’єкті управління (компанії) при веденні бізнесу, вибір шляхів її вирішення.

Підрозділ 1.1 «Бізнес-вимоги (Business Requirements)».

Бізнес-вимоги забезпечують основу та посилання для розробки всіх детальних вимог. Бізнес-вимоги можна отримати від вищого керівництва замовника або організації-розробника, виконавчого спонсора, менеджера застосунку, відділу маркетингу чи інших осіб, які мають чітке уявлення про те, навіщо проект здійснюється, і кінцеву цінність, яку він принесе надавати як компанії, так і клієнтам.

У пункті 1.1.1 «Обґрунтування (Background)» слід коротко викласти обґрунтування нового застосунку. Надайте загальний опис історії або ситуації, яка приводить до визнання того, що цей застосунок має бути створений.

У пункті 1.1.2 «Можливість для бізнесу (Business Opportunity)» опишіть існуючу ринкову можливість або бізнес-проблему, яка вирішується. Опишіть ринок, на якому буде конкурувати комерційний застосунок, або середовище, в якому використовуватиметься застосунок. Визначте проблеми, які зараз неможливо вирішити без застосунку, і те, як застосунок відповідає ринковим тенденціям або стратегічним напрямкам компанії.

У пункті 1.1.3 «Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)» опишіть важливі бізнес-цілі застосунку кількісно та вимірно. Цінність, надана клієнтам, буде описана в пункті 1.1.4, тому цей пункт має зосередитися на цінності, наданій бізнесу. Це може включати оцінку доходів або економії коштів, аналіз рентабельності інвестицій або цільові дати випуску. Визначте, як буде визначено та вимірювано успіх цього проекту, і опишіть фактори, які, ймовірно, матимуть найбільший вплив на досягнення цього успіху. Включайте речі, які безпосередньо контролюються організацією, а також зовнішні фактори. Встановіть вимірювані критерії, щоб оцінити, чи досягнуто бізнес-цілі.

Приклад.

Замовник:

• високі витрати на персонал;

• необхідність збільшення кількості відвідувачів.

У пункті 1.1.4 «Потреби клієнтів або ринку (Customer or Market Needs)» опишіть потреби типових клієнтів або сегментів ринку, включаючи потреби, які ще не задовольняються ринком або існуючими

системами. Опишіть проблеми, з якими зараз стикаються клієнти, які буде (або не буде) вирішувати новий застосунок, і як клієнти використовуватимуть застосунок. Визначте апаратне та програмне середовище клієнта, у якому повинен працювати застосунок. Визначте на високому рівні будь-який відомий критичний інтерфейс або вимоги до продуктивності. Уникайте включення будь-яких деталей дизайну чи реалізації. Надайте вимоги у вигляді пронумерованого списку, щоб можна було простежити до них більш детальні користувацькі або функціональні вимоги.

Приклад.

Клієнти:

1) Незручна процедура оформлення замовлення

2) Довге очікування

У пункті 1.1.5 «Бізнес-ризики (Business Risks)» узагальніть основні бізнес-ризики, пов’язані з розробкою цього застосунку, як-от конкуренція на ринку, проблеми з часом, прийняття користувачами, проблеми з впровадженням або можливий негативний вплив на бізнес. Оцініть серйозність ризиків і визначте будь-які дії щодо зменшення ризиків, які можна вжити.

Проілюструйте бізнес-вимоги за допомогою карти цілей (map of goals (Purpose - Direction - Function) (приклад наведено у дод. Г) та карти впливів (Impact Mapping) (приклад наведено у дод. Д).

Підрозділ 1.2 «Бачення рішення (Vision of the Solution)» визначає довгострокове бачення застосунку, який має бути створений для досягнення бізнес-цілей. Це бачення забезпечить контекст для прийняття рішень протягом усього життєвого циклу розробки застосунку. Бачення не повинно включати детальні функціональні вимоги або інформацію про планування проекту.

У пункті 1.2.1 «Заява про бачення (Vision Statement)» описується стисла заява про бачення, яка підсумовує мету та наміри нового застосунку та описує, що буде відбуватися, коли користувач буде використовувати цей застосунок. Заява про бачення має відображати збалансований погляд, який задовольнить потреби різноманітних клієнтів, а також потреби організації, що розвивається. Це може бути дещо ідеалістичним, але воно має ґрунтуватися на реаліях існуючих або

очікуваних ринків клієнтів, архітектурі підприємства, організаційних стратегічних напрямках, а також обмеженнях витрат і ресурсів (рис. 1).

Формула

<Назва застосунку/проєкту> це <категорія застосунку> для <цільова аудиторія> який <основне завдання> за допомогою <унікальність>.

Замовник планує заробляти/мати іншу вигоду за допомогою [спосіб монетизації].

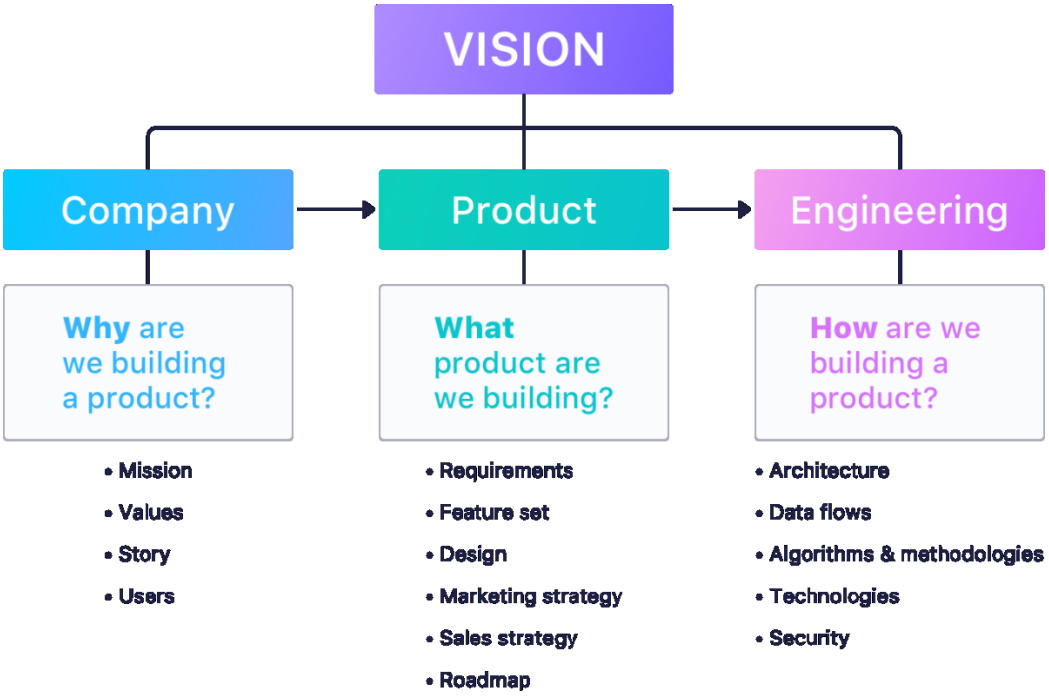

Рис. 1 Структура Заяви про бачення

У пункт 1.2.2 «Основні характеристики (Major Features)» включається пронумерований перелік основних характеристик нового застосунку, підкреслюючи ті характеристики, які відрізняють його від попередніх або конкуруючих застосунків. Конкретні вимоги користувача та функціональні вимоги можуть бути пов'язані з цими функціями.

У пункті 1.2.3 «Припущення та залежності (Assumptions and Dependencies)» вказуються будь-які припущення, зроблені під час розробки проєкту та написання цього документу про бачення та обсяг. Зверніть увагу на будь-які основні залежності, на які проєкт має покладатися для успіху, як-от певні технології, сторонні постачальники, партнери з розробки чи інші ділові відносини.

Підрозділ 1.3 «Сфера застосування та обмеження (Scope and Limitations)». Обсяг проекту визначає концепцію та діапазон запропонованого рішення. Також важливо визначити, що не входитиме в застосунок. Уточнення обсягу та обмежень допомагає встановити реалістичні очікування багатьох зацікавлених сторін. Він також забезпечує еталонну систему, за якою можна оцінити запропоновані функції та зміни вимог. Запропоновані вимоги, які виходять за рамки передбаченого застосунку, мають бути відхилені, якщо тільки вони не є настільки вигідними, що їх слід розширити (із супутніми змінами в бюджеті, графіку та/або ресурсах).

У пункті 1.3.1 «Обсяг початкового випуску застосунку (Scope of Initial Release)» описуються заплановані основні функції, які будуть включені в початковий випуск застосунку. Опишіть переваги, які застосунок має надати різним спільнотам споживачів, і загалом опишіть характеристики застосунку та характеристики якості, які дозволять йому надати ці переваги. Уникайте включення кожної можливої функції, яка колись може знадобитися будь-якій категорії потенційних клієнтів. Слід зосередитися на тих функціях і характеристиках застосунку, які забезпечать найбільшу цінність за найприйнятнішу вартість розробки для найширшого кола користувачів.

У пункті 1.3.2 «Обмеження та виключення (Limitations and Exclusions)» визначаються будь-які функції або характеристики застосунку, які зацікавлена сторона може передбачити, але які не планується включити в новий застосунок: фінансові, тимчасові, законодавчі, технічні.

У підрозділі 1.4 «Бізнес контекст (Business Context)» підсумовуються бізнес-проблеми, пов’язані з проєктом, включаючи профілі основних категорій клієнтів, припущення, які ввійшли в концепцію проєкту, і пріоритети управління проєктом.

Необхідно подати глосарій у вигляді таблиці (табл. 2).

Пункт 1.4.1 «Профілі зацікавлених сторін (Stakeholder Profiles)». Стейкхолдери – це особи, групи чи організації, які беруть активну участь у проекті, на яких впливає його результат або можуть вплинути на його результат. Профілі зацікавлених сторін ідентифікують клієнтів цього застосунку та інших зацікавлених сторін, а також зазначають їхні основні інтереси в застосунку. Охарактеризуйте клієнтів бізнес-рівня, цільові

ринкові сегменти та різні класи користувачів, щоб зменшити ймовірність появи неочікуваних вимог, які не можуть бути задоволені через обмеження графіка чи масштабу. Для кожної категорії зацікавлених сторін профіль включає основну цінність або переваги, які вони отримають від застосунку, їхнє ймовірне ставлення до застосунку, основні функції та характеристики, що цікавлять, і будь-які відомі обмеження, які необхідно врахувати.

Таблиця 2

Глосарій

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
2. Користувачі системи	

Приклад глосарію наведено у дод. Ж.

Приклади цінностей для зацікавлених сторін:

- підвищення активності;
- скорочена обробка інформації;
- економія коштів;
- налагоджені бізнес-процеси;
- автоматизація завдань, які раніше виконувалися вручну;
- здатність виконувати абсолютно нові завдання чи функції;
- відповідність чинним стандартам або правилам;
- покращена зручність використання або зменшений рівень розчарування порівняно з поточними програмами.

Приклад наведення профілей зацікавлених сторін наведено у дод. З.

У пункті 1.4.2 «Операційне середовище (Operating Environment)» опишіть середовище, у якому використовуватиметься система, і визначте основні вимоги до доступності, надійності, застосовуваності та

цілісності. Ця інформація суттєво вплине на визначення архітектури системи. Розгляньте такі питання, як:

• Чи географічно розподілені користувачі чи розташовані близько один до одного? У скількох часових поясах вони знаходяться?

• Коли користувачам у різних місцях потрібен доступ до системи?

• Де генеруються та використовуються дані? Як далеко одне від одного знаходяться ці місця? Чи потрібно об'єднувати дані з кількох місць?

• Чи відомий конкретний максимальний час відповіді для доступу до даних, які можуть зберігатися віддалено?

• Чи можуть користувачі терпіти перебої в обслуговуванні чи безперервний доступ до системи є критичним для роботи їхнього бізнесу?

• Які засоби контролю безпеки доступу та вимоги до захисту даних необхідні?

Метою другого розділу «Специфікація вимог до модуля (системи)» є розроблення і детальна специфікація вимог до модуля (системи), що розробляється.

Специфікація вимог до модуля (системи) (SRS) повинна містити достатньо інформації для того, щоб розробники могли завершити створення описаного програмного забезпечення. Вона не тільки містить опис програмного забезпечення, що розробляється, але й мету, якій воно буде служити: що повинно робити програмне забезпечення і як воно повинно працювати.

Документ SRS, як правило, включає такі елементи:

- мета програмного забезпечення, що розробляється;
- загальний опис програмного забезпечення;
- функціональність програмного забезпечення або те, що воно повинно робити;
- застосунковість програмного забезпечення у виробничій ситуації;
- нефункціональні вимоги;
- зовнішні інтерфейси або те, як програмне забезпечення буде взаємодіяти з обладнанням або іншим програмним забезпеченням, до якого воно повинно підключатися;

- обмеження проєктування або обмеження середовища, в якому буде працювати програмне забезпечення.

У підрозділі 2.1 «Аналіз існуючих рішень (аналогів, конкурентів)» має бути наведена коротка характеристика аналогів / конкурентів (не менше 3-5).

Порівняльний аналіз має проводитися за критеріями (табл. 3).

Таблиця 3

Порівняльний аналіз існуючих рішень за критеріями

	Аналог 1	Аналог 2	…	Аналог 5
Критерій 1				
Критерій 2				
…				
Критерій 6				

Порівняльний аналіз має завершитися висновком щодо переваг / недоліків, а також необхідності розробки власного рішення.

Підрозділ 2.2 «Специфікація функціональних вимог». Функціональні вимоги до системи, що розробляється – це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система. На відміну від нефункціональних вимог, які визначають якою система повинна бути, функціональні вимоги визначають, що система повинна робити.

Розроблення варіантів використання містить:

- діаграму варіантів використання;
- специфікацію варіантів використання.

Діаграма варіантів використання відображає функціональність, яка буде реалізована в програмному застосунку. Варіант використання можна розглядати як функцію, що реалізовується системою. Однак будь-яка функція повинна мати цінність і давати можливість отримати кінцевий результат для кінцевого користувача застосунку або послуги. Тому при специфікації варіанта використання серед усієї функціональності системи виділяють лише ту функціональність, яка:

- корисна конкретному кінцевому користувачеві;

- дозволяє отримувати користувачеві конкретні закінчені результати.

Специфікація варіантів використання.

Кожен варіант використання повинен мати опис. У проекті слід навести опис варіантів використання, що реалізують основну функціональність (зазвичай крім ведення довідників) у вигляді таблиці (табл. 4).

Таблиця 4

Специфікація варіантів використання

Назва варіанта використання	
Короткий опис	
Дійові особи	
Передумова	
Основний потік подій (сценарій)	1. 2. ...
Альтернативний потік подій	1. 2. ...
Постумова	
Виключення	

До кожного варіанту використання слід надати діаграму діяльності (Activity).

Підрозділ 2.3 «Нефункціональні вимоги» має містити:

- вимоги продуктивності;
- вимоги надійності;
- вимоги безпеки;
- атрибути якості програмного застосунку;
- вимоги інтернаціоналізації;
- вимоги бази даних.

У підрозділі 2.4 «Проектування інтерфейсу користувача (User Interfaces)» треба розробити графічний інтерфейс користувача з застосунком у вигляді каркасу (wireframe) або макету (mockup), які використовуються для отримання схвалення зацікавлених осіб, щодо пропонованої концепції.

В цьому підрозділі потрібно описати:

1. Логіку між користувачами та програмним забезпеченням. Слід приділити увагу детальному опису від зображень на екрані до макетів програм, від вмісту меню до звітів тощо.

2. Способи оптимізації інтерфейсу. Слід перерахувати важливі рекомендації та обмеження для оптимізації інтерфейсу користувача.

Далі слід описати будь-які інтерфейси зв'язку з іншими системами або пристроями, такими як локальні мережі, віддалені послідовні пристрої тощо. Описаний проект має підтримувати всі типи веб-браузерів.

Для візуалізації розроблюваного програмного застосунку слід додавати до SRS каркаси програми, макети та прототипи (рис. 2).

• Каркас (wireframe) – це зображення конструкції з низьким рівнем деталізації. Зазвичай це чорно-білий план сторінки застосунку. У ньому намічено розташування основних елементів: кнопок, зображень, текстів. Він не виконує жодних реальних функцій сайту. Результати взаємодій, кліків, анімацій необхідно описувати додатково в коментарях. Каркас використовують, щоб визначити, де і який контент буде розміщено.

• Макет (mockup) – це детальне статичне представлення дизайну. Він включає кольори, зображення, типографіку. Макет більше схожий на гарне зображення програми або веб-сайту. Макет використовують, щоб створити стиль проекту, представити та узгодити візуальні деталі із замовником.

• Прототип (prototype) – це імітація взаємодії користувача з інтерфейсом застосунку. Це інтерактивна версія каркасу. Прототип використовують, щоб проілюструвати розташування блоків і кнопок і провести тестування зручності використання.

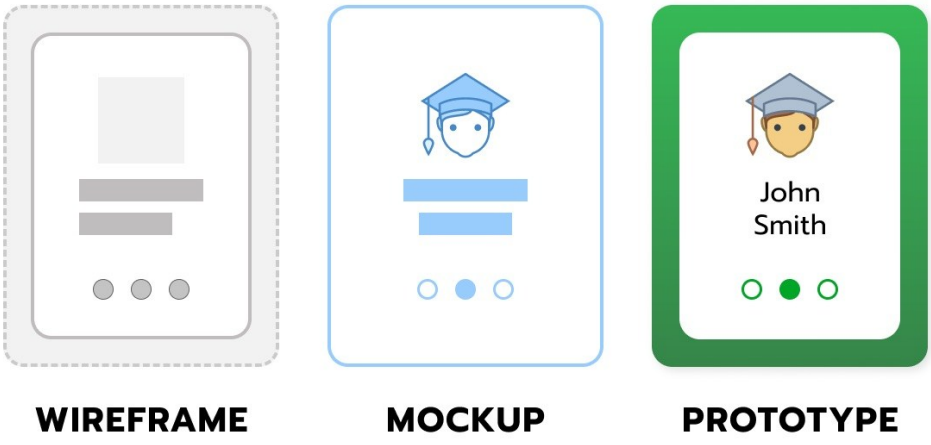

Рис. 2 Приклади каркасу, макету, прототипу

Метою проектування інтерфейсу є отримання ранньої реакції користувачів на пропоновану концепцію системи за допомогою некоштовних засобів. У якості інструментальних засобів використовуються Marvel, Pencil Project, Use Your Interface, UX Myths, Mobile Patterns, Good UI, Microsoft Visio, Microsoft PowerPoint тощо.

Підрозділ 2.5 «Деталізація нефункціональних вимог. Логічна структура даних (Detailed Non-Functional Requirements. Logical Structure of the Data)». Проектування структури даних застосунку включає проектування бази даних і проектування діаграми класів. Ці два підходи необхідні, оскільки застосунок повинен зберігати дані.

Отже, обговорення проектування системи буде включати в себе діаграму "сутність-зв'язок" (ER) та діаграму класів.

Після визначення вимог до застосунку, необхідно визначити різні сутності, що беруть участь в ньому. По-перше, це логічна ER діаграма - діаграма "сутність-зв'язок“, в якій сутності повинні бути нормалізовані для встановлення належних взаємозв'язків. Також слід зробити детальний опис таблиць бази даних для системи, яка розробляється.

Далі слід навести UML-діаграму класів (Class Diagram), що реалізують основну бізнес-логіку програмної системи, та її опис.

Приклади логічної ER діаграми, опис таблиць бази даних та UML-діаграми класів наведені у дод. К.

Висновки до роботи – це резюме за результатами всієї роботи. Ця частина має особливу важливість, оскільки тут повинні бути наведені підсумкові результати роботи.

У пояснювальній записці має міститися висновки по кожному етапу виконаного проекту та по роботі в цілому, які необхідно співвідносити з метою і завданням на курсове проектування.

Необхідно зазначити практичну цінність результатів роботи, дати рекомендації для подальшого вдосконалення об’єкту проектування. Зазначаючи практичну цінність одержаних результатів, важливо окреслити ступінь їх готовності до використання, масштабів використання.

Список використаних джерел повинен містити відомості про літературні джерела, використані при розробленні проекту.

Список літератури – це реєстр використаних джерел за темою проекту у якнайширшому значенні. Тому не слід обмежуватися лише цитованою літературою. У список варто включати усі матеріали, які були прочитані, переглянуті, проаналізовані при роботі над курсовим проектом і стосуються його теми. Бажано виявляти джерела якомога повніше, пам'ятаючи, що бібліографічний список до проекту – це підсумок вивчення проблеми і передумова подальших наукових досліджень.

Список використаних джерел повинен містити не менше 30 джерел. Він подається мовою оригіналу, розміщується в алфавітному порядку прізвищ перших авторів або назв, та нумерується в порядку їх зростання. Нумерація безперервна.

Роботи одного автора розташовані за алфавітом назв або в хронології їх написання. Алфавітний список розташований за алфавітом у такій послідовності:

література зведеного кириличного алфавіту, для джерел на мовах з кириличною графікою (російська, українська, болгарська та ін.);

література в латинському алфавіті,

електронні ресурси в тій же послідовності що й друковані видання (спочатку кирилицею, а потім латиницею).

Список літературних джерел обов’язково повинен містити прізвище та ініціали автора, повну назву джерела, місто видавництва, видавництво та рік видання, кількість сторінок чи посилання на сторінки тощо. Загальний обсяг книги в сторінках вказується, якщо посилання на неї проводиться повністю, сторінки (від... до) відмічаються, якщо посилання відносяться до окремої частини літературного джерела.

У Додатках вміщують матеріал, який є необхідним для повноти пояснювальної записки, але не може бути послідовно розміщений в її основній частині через великий обсяг або з інших причин.

Ілюстрації (діаграми бізнес-процесів, сценарії діалогів та ін.), таблиці, проміжні математичні докази, формули та розрахунки, текст допоміжного характеру та ін. можуть бути оформлені у вигляді додатків.

5. Вимоги до оформлення пояснювальної записки курсового проєкту

Важливе значення при роботі над курсовим проєктом має його оформлення, до якого пред‘являються певні вимоги.

При оформленні тексту пояснювальної записки слід керуватися Методичними рекомендаціями до оформлення звітів, курсових проєктів та дипломних робіт (проєктів) [1].

Рекомендована література

Основна

1. Методичні рекомендації до оформлення звітів, курсових проєктів та дипломних робіт (проєктів) для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи і технології" [Електронний ресурс] / уклад. І. О. Ушакова, Г. О. Плеханова, О. М. Беседовський. – Харків : ХНЕУ ім. С. Кузнеця, 2021. – 45 с. - Режим доступу: http://repository.hneu.edu.ua/handle/123456789/27413

Додаткова

2. Інженерія програмного забезпечення: робоча програма навчальної дисципліни для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня : [Електронне видання] / уклад. І.О. Золотарьова. – Харків : ХНЕУ ім. С. Кузнеця, 2024. – 13 с. - Режим доступу: http://repository.hneu.edu.ua/handle/123456789/ 32174

3. Бородкіна І.Л., Борожкін Г.О. Інженерія програмного забезпечення. – Центр учбової літератури, 2020. – 204 с.

Ресурси мережі Internet

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. Персональна навчальна система "Курсовий проєкт "Інженерія

програмного забезпечення" (6.121.010)" [Електронний ресурс]. - Режим доступу: https://pns.hneu.edu.ua/course/view.php?id=6597

Додатки

Додаток А

Перелік рекомендованих тем курсових проєктів

Розумні міста

1. Система моніторингу та оптимізації вуличного освітлення:
   - Розробка програмного забезпечення для збору даних про освітленість вулиць, датчики руху, та прогнозування потреб у освітленні.
   - Створення алгоритмів для динамічного керування освітленням з метою економії енергії та покращення видимості.
2. Система моніторингу та управління паркувальними місцями:
   - Розробка мобільного додатку для пошуку вільних паркувальних місць у реальному часі.
   - Створення системи датчиків для відстеження зайнятості паркувальних місць.
3. Система моніторингу та прогнозування забруднення повітря:
   - Розробка програмного забезпечення для збору даних про якість повітря з датчиків, розташованих по місту.
   - Створення алгоритмів для прогнозування забруднення повітря та оповіщення жителів.
4. Система моніторингу та управління муніципальними відходами:
   - Розробка мобільного додатку для повідомлення про переповнені сміттєві баки та замовлення вивезення сміття.
   - Створення системи датчиків для моніторингу рівня заповнення сміттєвих баків.

Карта бомбосховищ

5. Мобільний додаток з картою бомбосховищ:
   - Розробка мобільного додатку з інтерактивною картою бомбосховищ, що включає інформацію про їх розташування, місткість, доступність та інші характеристики.
   - Додавання функції пошуку найближчого бомбосховища та прокладання маршруту до нього.

6. Веб-сайт з картою бомбосховищ:
    - Створення веб-сайту з картою бомбосховищ, що містить інформацію про їх розташування, місткість, доступність та інші характеристики.
    - Додавання функції пошуку бомбосховищ за адресою, районом або іншими критеріями.
7. Система онлайн-реєстрації для отримання соціальних послуг:
    - Розробка програмного забезпечення для онлайн-реєстрації людей, які потребують соціальних послуг.
    - Створення алгоритмів для автоматизованого розподілу ресурсів та підтримки зв'язку з людьми, які потребують допомоги.
8. Система моніторингу та підтримки людей похилого віку:
    - Розробка мобільного додатку для моніторингу стану здоров'я та життєдіяльності людей похилого віку.
    - Створення системи датчиків для відстеження показників здоров'я та повідомлення про надзвичайні ситуації.
9. Система онлайн-навчання для людей з обмеженими можливостями:
    - Розробка веб-платформи з онлайн-курсами та інтерактивними матеріалами для людей з обмеженими можливостями.
    - Створення адаптивного інтерфейсу та доступного контенту для людей з різними типами інвалідності.
10. Система онлайн-платформи для волонтерів:
    - Розробка веб-платформи для з'єднання волонтерів з людьми, які потребують допомоги.
    - Створення алгоритмів для підбору волонтерів на основі їхніх навичок та досвіду.

Інші теми

11. Система моніторингу та управління водними ресурсами міста
12. Система раннього попередження про стихійні лиха
13. Система онлайн-голосування для участі в місцевому самоврядуванні
14. Система моніторингу та покращення доступності громадського

простору

Закінчення додатку А

15. Система онлайн-платформи для обміну знаннями та досвідом
16. Система онлайн-платформи для збору коштів на соціальні проекти
17. Система моніторингу та покращення якості життя в громаді

Додаток Б

Зразок оформлення титульного аркуша

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Пояснювальна записка

до курсового проєкту з “Інженерії програмного забезпечення”
на тему: «Проєтування модуля “Назва предметної області”»

Здобувача(ки) 3 курсу 6.04.121.010.21.01 групи
Спеціальності 121 "Інженерія програмного забезпечення"
першого (бакалаврського) рівня
Вареник Г.В.
Керівник: професор кафедри ІС, к.е.н., доцент,
Золотарьова І.О.

Національна шкала _____

Кількість балів: _____Оцінка: ECTS _____

Члени комісії

_____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

м. Харків – 202_ рік

Додаток В

Зразок оформлення змісту пояснювальної записки

ЗМІСТ

Вступ……6
1 Бізнес-аналіз предметної області <назва предметної області>…… 7
1.1 Бізнес-вимоги (Business Requirements) …… ……7
1.1.1 Обґрунтування (Background)……..……8
1.1.2 Можливість для бізнесу (Business Opportunity)……..……9
1.1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)……..10
1.1.4 Потреби клієнтів або ринку (Customer or Market Needs)……....11
1.1.5 Бізнес-ризики (Business Risks……...12
1.2 Бачення рішення (Vision of the Solution)……13
1.2.1 Заява про бачення (Vision Statement)……..……..13
1.2.2 Основні характеристики (Major Features)……..……14
1.2.3 Припущення та залежності (Assumptions and Dependencies)…..…14
1.3 Сфера застосування та обмеження (Scope and Limitations)……15
1.3.1 Обсяг початкового випуску застосунку (Scope of Initial Release)….15
1.3.2 Обмеження та виключення (Limitations and Exclusions)……....15
1.4 Бізнес контекст (Business Context)……..……16
1.4.1 Профілі зацікавлених сторін (Stakeholder Profiles)……..……16
1.4.2 Операційне середовище (Operating Environment)……..…17
2 Специфікація вимог до модуля (системи)……..……18
2.1 Аналіз існуючих рішень (аналогів, конкурентів)……..…18
2.2 Специфікація функціональних вимог……..21
2.3 Нефункціональні вимоги……..……24
2.4 Проектування інтерфейсу користувача (User interfaces)……..27
2.5 Деталізація нефункціональних вимог. Логічна структура даних (Detailed Non-Functional Requirements. Logical Structure of the Data)……28
Висновки……...30
Список використаних джерел……..31
Додатки……32

Додаток Г

Приклад карти цілей

Рис. Г.1 Аналіз вимог бізнесу в формі карти цілей

Додаток Д

Приклад карти впливів

Таблиця Д.1

Аналіз вимог бізнесу в формі карти впливів

Навіщо (мета)	**Хто (Дійові особи)**	**Як (Вплив, дія)**	**Що (задачі, вимоги)**
Зменшити витрати на утримання офіціантів на X	Менеджер	Керувати електронним меню	Керувати базою даних доступних страв
			Перевіряти та підтверджувати замовлення
	Офіціант	Зменшити обсяг виконуваної роботи	Займатися тільки подачею страв та отриманням оплати
	Замовник	Не звертатися до офіціантів для замовлення	Використовувати електронне меню
Збільшити кількість сучасних клієнтів на N	PR-менеджер	Провести PR-кампанію введення електронного меню	Зробити СМС та email розсилку постійним клієнтам
			Купити рекламу у соціальних мережах

Додаток Ж

Приклад глосарію

Таблиця Ж.1

Глосарій

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
Електронне меню	Веб-додаток, який має доступ до бази даних доступних для замовлення товарів та надає користувачу функціонал для виконання замовлення.
Замовлення	Список страв, які обрав замовник.
2. Користувачі застосунку	
Замовник	Людина, яка має доступ в систему для виконання замовлення.
Менеджер	Людина, яка має доступ в систему з особливими правами для керування інформацією.

Додаток 3

Приклад профілей зацікавлених сторін

Таблиця 3.1

Профілі зацікавлених сторін

Класи користувачів	**Характеристики користувачів**
Менеджер	Менеджер – це користувач, який керує базою замовлень. *Список можливостей:* 1. Додавання інформації про нові страви. 2. Редагування та видалення інформації про існуючі страви. 3. Підтвердження замовлень.
Офіціант	Офіціант – це користувач, який відносить готове замовлення замовнику. *Список можливостей:* 1. Отримання інформації про місцезнаходження замовника 2. Віднесення готових страв замовнику 3. Прийняття оплати за замовлення
Замовник	Замовник – це користувач, який робить замовлення на певні страви. *Список можливостей:* 1. Ознайомлення зі списком доступних для замовлення страв 2. Виконання замовлення 3. Проведення оплати за замовлення

Додаток К

Приклад оформлення підрозділу з проєктування бази даних

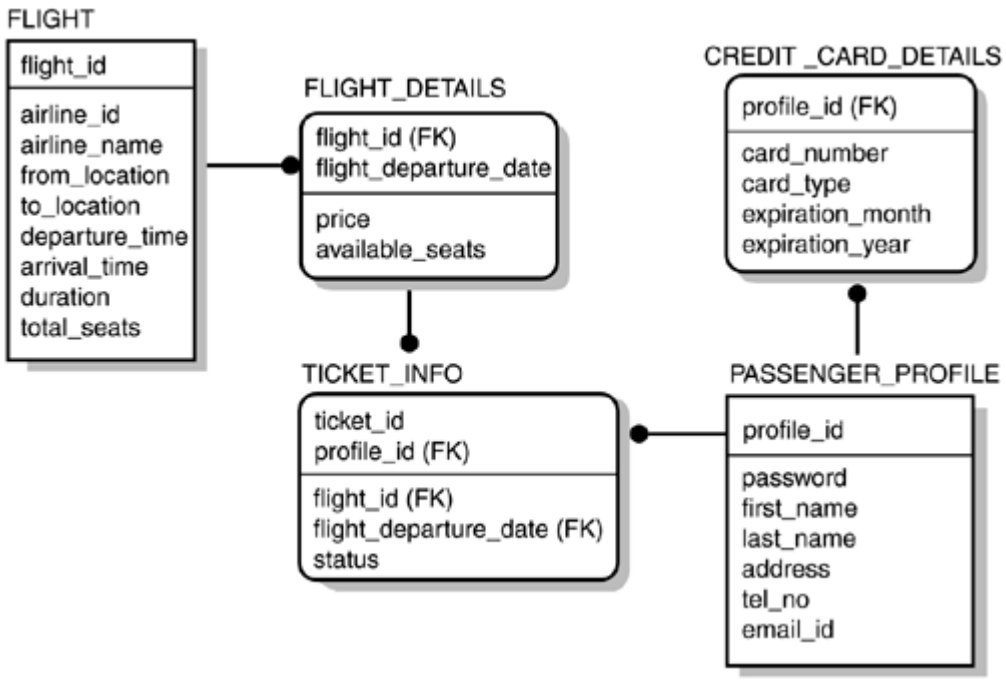

Рис. К.1 Логічна модель БД

Таблиця К.1

Структура таблиці «FLIGHT»

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.

Таблиця К.2

Структура таблиці «FLIGHT_DETAILS»

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.

Таблиця К.3

Структура таблиці «PASSENGER_PROFILE»

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.

Таблиця К.4

Структура таблиці «TICKET_INFO»

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.

Таблиця К.5

Структура таблиці «CREDIT_CARD_DETAILS»

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.

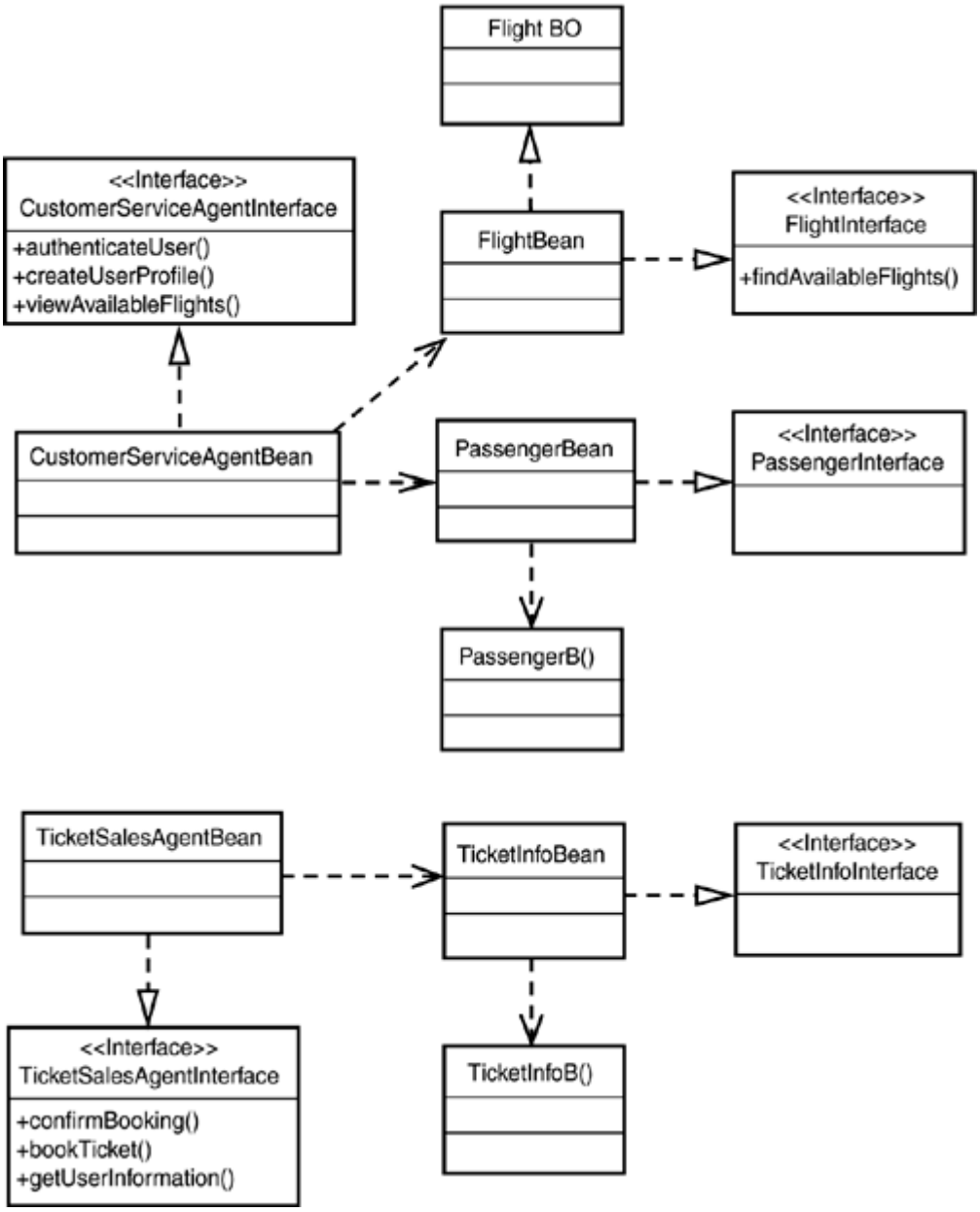

Рис. К.2 UML-діаграма класів

Зміст

Вступ ……		3
1.	Мета й завдання курсового проєктування ……..	4
2.	Організація курсового проєктування …….....	5
3.	Структура та обсяг курсового проєкту …….	6
4.	Методичні рекомендації до розроблення структурних елементів пояснювальної записки курсового проєкту …….	7
5.	Вимоги до оформлення пояснювальної записки курсового проєкту …….	20
	Рекомендована література …….	21
	Додатки …….....	23

НАВЧАЛЬНЕ ВИДАННЯ

ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до виконання курсового проєкту для здобувачів вищої освіти спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня

Укладачі: Золотарьова Ірина Олександрівна
Беседовський Олексій Миколайович

Відповідальний за випуск Бондаренко Д. О.

Редактор
Коректор

План 2024 р. Поз. № 81 ЕВ .
Підп. до друку Формат 60 х 90 1/16. Папір MultiCopy. Друк Riso.
Ум.-друк. арк. Обл.-вид. арк. Тираж прим. Зам. №
Видавець і виготівник — видавництво ХНЕУ ім. С. Кузнеця, 61166, м. Харків, пр. Науки, 9-А
Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи ***Дк № 481 від 13.06.2001 р.***