

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація / освітня програма 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки
Криворучко О. В.
"7" листопада 2019 р.

Завдання на випускна кваліфікаційна робота студентів

Руденко Андрій Олександрович

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи Розробка online-платіжної системи

Затверджена наказом ректора від «02» грудня 2019 р. № 4112

2. Строк здачі студентом закінченої роботи

3. Цільова установка та вихідні дані до роботи

Мета роботи Розробка програмного забезпечення мовою програмування Java за допомоги Spring Framework MVC

Об'єкт дослідження Фінансові транзакції в платіжних системах

Предмет дослідження Технологія розробки веб-додатку на основі об'єктно орієнтованій мові програмування Java з використанням Spring Framework

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП.....

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ПЛАТІЖНОЇ СИСТЕМИ
ТА WEB-ДОДАТКА.....

1.1 Введення в предметну область.....

1.2 Об'єктно-орієнтована мова програмування Java

1.3 Фреймворк Spring MVC для Java-платформи

1.4 Формування технічного завдання

1.5 Висновок до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Інтегроване середовище розробки програмного забезпечення для мови
програмування Java.....

2.2 Розробка архітектури та аналіз структури додатку

2.3 Проектування структури реляційної бази даних.....
та її реалізація засобами MySQL.....

2.4 Моделювання графічного інтерфейсу додатку

2.5 Висновки до розділу 2

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ON-LINE ПЛАТІЖНОЇ СИСТЕМИ...

3.1 Графічний інтерфейс розробленого додатку.....

3.2 Архітектура розробленого додатку та інструкція користувача.....

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ.....

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>		
2.	<i>Вступ та перелік літературних джерел</i>		
3.	<i>Розділ 1. «Теоретичні відомості розробки платіжної системи та web-додатка»</i>		
4.	<i>Розділ 2. «Проектування програмного забезпечення»</i>		
5.	<i>Розділ 3. «Програмна реалізація on-line платіжної системи»</i>		
6.	<i>Висновки</i>		
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>		
8.	<i>Підготовка автореферату та презентації доповіді</i>		
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>		
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>		
11.	<i>Здача прожитої випускної кваліфікаційної роботи на кафедру</i>		
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «__» _____ 20__ р.

8. Науковий керівник випускної кваліфікаційної роботи

Цензура М. О.

9. Гарант освітньої програми

Цензура М. О.

10. Завдання прийняв до виконання студент

Руденко А. О.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(nidnuc, dama)

Цензура М. О.

(ПИБ, підпис, дата)

Випускна кваліфікаційна робота студента Руденко А.О. може бути допущена до захисту екзаменаційній комісії.

Цензура М. О.

Криворучко О. В.

« » 20 p.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (50с., 14 рис., 5 таб., 8 додатків).

Об'єкт розробки – веб-додаток, online-платіжна система, для користувачів всесвітньої мережі Internet з миттєвими транзакціями та відсутніми комісіями.

Завданням дослідження є:

- Аналіз наявності інструментів для розробки веб-додатку;
- Формування технічного завдання для розробки online-платіжної системи;
- Розробка архітектуру та аналіз структури веб-додатку;
- Проектування структури бази даних та реалізація засобами MySQL
- Моделювання інтерфейсу веб-додатку для користувачів;
- Розробка платіжної системи.

В процесі розробки було використано мову програмування Java та Spring Framework. IntelliJ IDE – для розробки графічного інтерфейсу та створення веб-додатку.

Ключові слова: веб-додаток, об'єктно-орієнтоване програмування, розробка графічного інтерфейсу, Java, IntelliJ IDE, Spring Framework, Spring MVC, Spring DATA JPA, Spring Security, DB MySQL, ORM, UI, online-платіжна система, архітектура, база даних, гарантії, користувачі, транзакції.

ANNOTATION

Qualification work includes an explanatory note (50 pages, 14 figures, 5 tables, 8 appendices).

The object of development is a web application, an online payment system, for users of the World Wide Web with instant transactions and no fees.

The task of the study is:

- Analysis of the availability of tools for web application development;
- Formation of terms of reference for the development of online payment system;
- Development of architecture and analysis of the structure of the web application;
- Design of database structure and implementation by means of MySQL
- Simulation of the web application interface for users;
- Development of a payment system.

The Java and Spring Framework programming languages were used in the development process. IntelliJ IDE - for developing a graphical interface and creating a web application.

Keywords: web application, object-oriented programming, graphical interface development, Java, IntelliJ IDE, Spring Framework, Spring MVC, Spring DATA JPA, Spring Security, DB MySQL, ORM, UI, online payment system, architecture, database, wallets, users, transactions.

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка online-платіжної системи

Студента 4 курсу, 6 групи,
спеціальності «Інженерія
програмного забезпечення»

підпис студента

Руденка Андрія
Олександровича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2020

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ПЛАТІЖНОЇ СИСТЕМИ ТА WEB-ДОДАТКА	7
1.1 Введення в предметну область.....	7
1.2 Об'єктно-орієнтована мова програмування Java	8
1.3 Фреймворк Spring MVC для Java-платформи	10
1.4 Формування технічного завдання	15
1.5 Висновок до розділу 1.....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	17
2.1 Інтегроване середовище розробки програмного забезпечення для мови програмування Java.....	17
2.2 Розробка архітектури та аналіз структури додатку.....	19
2.3 Проектування структури реляційної бази даних.....	24
та її реалізація засобами MySQL.....	24
2.4 Моделювання графічного інтерфейсу додатку	28
2.5 Висновки до розділу 2	34
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ON-LINE ПЛАТІЖНОЇ СИСТЕМИ	35
3.1 Графічний інтерфейс розробленого додатку.....	35

					КНТЕУ 121 06-244.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка online-платіжної системи	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				Зміст	2	47
Керівник		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розробив		Руденко А.О			Зміст			

3.2 Архітектура розробленого додатку та інструкція користувача.....	38
3.3. Висновки до розділу 3	39
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
ДОДАТКИ	43
ДОДАТОК А	43
ДОДАТОК Б.....	44
ДОДАТОК В.....	45
ДОДАТОК Г.....	46
ДОДАТОК Д.....	47
ДОДАТОК Е.....	48
ДОДАТОК Ж.....	49
ДОДАТОК З	50

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум. _	Підпис	Дата		3

ВСТУП

Інноваційні технології підкорюють нові вершини і розвиваються неймовірними темпами. Технологія чи рішення, в будь-якому виді, націлена на вирішення потреби клієнта. Зазвичай програмні продукти виконують автоматизовані та систематизовані функції для мінімізації витрати часу користувача та вирішення конкретних задач.

Разом із розвитком інноваційних технологій розвиваються мови програмування. З'являється багато можливостей для реалізації ідей та нестандартних рішень, що все ближче наближає до вирішення конкретних задач користувачів.

Але потреби користувача у стартапах чи великих компаніях не стоять на першому місці і тому ринок переповнений великою кількістю пропозицій, що розпиляє увагу користувача з невеликою ймовірністю обрати продукт, який дозволить назавжди вирішити проблему з життя користувачів.

Для створення успішного програмного продукту потрібно проаналізувати ринок, виявити потребу цільової аудиторії та закрити потребу інноваційним рішенням, котре допоможе кожному користувачу при використанні покращити та облегшити процеси.

Найпопулярнішими методами все більш становляться веб-додатки, що дозволяють аудиторії використовувати програмні рішення з будь-якого пристрою маючи лише підключення до інтернету.

Інновації не пройшли повз фінансову індустрію і все більше програмних рішень з'являються у зв'язку з потребами користувачів.

З появою та швидким зростанням кількості кінцевих користувачів інтернету люди почали використовувати online-сервіси з метою купівлі

					КНТЕУ 121 06-24.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка online-платіжної системи	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В				В	4	47
Керівник		Цензура М.О.						
Гарант		Цензура М.О.						
Розробив		Руденко А.О			Вступ	Факультет інформаційних технологій, 4 курс, 6 група		

товару чи отримання платних послуг, що дозволило розвиватися ринку електронної комерції.

Фінансові операції через всесвітню мережу інтернет це новий тренд в усьому світі. Найпоширенішими та найзручнішими інструментами для проведення фінансових операції є online-платіжні системи. Але більшість платіжних систем мають низьку швидкість проведення фінансових операцій та велику комісію, що не дозволяє багатьом бізнесам чи користувачам остаточно перейти на онлайн методи фінансових операцій.

Актуальність даної теми полягає в тому, що незважаючи на велику кількість online-платіжних систем, у сфері електронної комерції досі немає аналогів платіжних сервісів з миттєвими фінансовими транзакціями та відсутніми комісіями.

Тому було вирішено розробити веб-додаток - online-платіжну систему на основі об'єктно орієнтованій мові програмування Java з використанням Spring Framework MVC.

Метою дослідження випускної кваліфікаційної роботи є розробка програмного забезпечення мовою програмування Java за допомоги Spring Framework MVC.

Об'єктом дослідження є фінансові транзакції в платіжних системах.

Предметом – технологія розробки веб-додатку на основі об'єктно орієнтованій мові програмування Java з використанням Spring Framework.

Завдання дослідження:

- Аналіз наявності інструментів для розробки веб-додатку;
- Формування технічного завдання для розробки online-платіжної системи;
- Розробка архітектуру та аналіз структури веб-додатку;
- Проектування структури бази даних та реалізація засобами MySQL

Зм.	Лист	№ Докум.	Підпис	Дата

КНТЕУ 121 06-24.БР

Лист
5

- Моделювання інтерфейсу веб-додатку для користувачів;
- Розробка платіжної системи.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		6

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ВІДОМОСТІ РОЗРОБКИ ПЛАТІЖНОЇ СИСТЕМИ ТА WEB-ДОДАТКА

1.1 Введення в предметну область

Online-платіжні системи – призначені для здійснення платіжних операцій в інтернеті. За допомогою платіжної системи можна здійснювати розрахунок за товари та послуги. Всі фінансові операції здійснюються за допомогою балансу користувача, який складається з електронних грошей.

Електронні гроші – це аналог готівки, яка оцифровується і відображається на балансі користувача для змоги проведення подальших фінансових операцій в інтернеті.

Фінансові операції – це рух та обмін електронних грошей на балансі користувачів при взаємній згоді.

Виникнення online-платіжних систем зумовило те, що електронна комерція в світі стала набирати значних обертів, а розрахунки кредитними картками стали небезпечними та повільними.

Цілі сервісу електронних платежів:

1. Забезпечення простого і швидкого процесу відкриття електронного рахунку без обмежень в кількості створених облікових записів.
2. Простота і велика швидкість проведення будь-яких фінансових операцій.
3. Зменшення або відсутність комісій за фінансові операції в незалежності від сум або кількості проведених транзакцій.
4. Забезпечення конфіденційності та анонімності без доступу до інформації користувача третім особам чи організаціям.

					КНТЕУ 121 06-24.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка online-платіжної системи	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В				Р1	7	47
Керівник		Цензура М.О.						
Гарант		Цензура М.О.						
Розробив		Руденко А.О				Теоретичні відомості розробки платіжної системи та web-додатка	Факультет інформаційних технологій, 4 курс, 6 група	

За останніх 5 років найпопулярнішими методами забезпечення безпеки фінансових операцій користувачів стали методи шифрування за допомогою різних алгоритмів. Інтегрування методів шифрування в платіжні системи вимагає збільшення обсягу використання ресурсів.

1.2 Об'єктно-орієнтована мова програмування Java

Java - строго типізована об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems та в подальшому придбана компанією Oracle.

Створення Java - це дійсно один з найзначніших кроків вперед в області розробки програмного середовища. Мова Java потрібна для якісного стрибка в створенні веб-додатків для мережі Інтернет.

Три ключових елементи об'єдналися в технології мови Java і виділили її з всіх існуючих мов програмування:

1. Мова Java реалізує потужність об'єктно-орієнтованої розробки додатків, поєднуючи простий і знайомий синтаксис з надійним і зручним в роботі середовищем розробки. Це дозволяє широкому кругу програмістів швидко створювати нові програми.
2. Мова Java надає програмісту багатий набір класів об'єктів для чіткого абстрагування багатьох системних функцій, використовуваних при роботі з вікнами, мережею і для введення-виводу інформації.
3. Ключова риса цих класів полягає в тім, що вони забезпечують створення незалежних від платформ абстракцій для широкого спектра системних інтерфейсів.

Мова Java зародилася як частина проекту створення програмного забезпечення для різних побутових приладів. Реалізація проекту була почата на мові C++, але незабаром виник ряд проблем, найкращим засобом боротьби з якими була зміна самого інструмента - мови програмування. Стало

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		8

очевидним, що необхідна кросплатформена мова програмування, що дозволяє створювати програми, які не доводилося б компілювати окремо для кожної архітектури і можна було б використовувати на різних пристроях з різними операційними системами.

Можливість вирішення завдань будь-яких рівнів складності та взаємодії з користувачем існує лише в тому випадку, коли використовувана для розробки платформа надає повнофункціональну середу програмування.

У наші дні існує багато систем програмування, які пишаються тим, що в них реалізована можливість досягати мету різними способами. У мові Java прагнення до простоти часто приводило до створення неефективних і невиразних мов типу командних інтерпретаторів. Java до числа таких мов не відноситься.

У популярній літературі популярною темою є обговорення питань безпеки. Один з ключових принципів розробки мови Java полягав у забезпеченні захисту від несанкціонованого доступу. Програми на Java не можуть викликати глобальні функції і отримувати доступ до довільних системних ресурсів, що забезпечує в Java рівень безпеки, недоступний для інших мов.

Java обмежує програміста в декількох ключових областях і таким чином сприяє виявленню помилок на ранніх стадіях розробки програми. У той же час в ній відсутня велика кількість помилок, властивих іншим мовам програмування.

Java створювалася як засіб, який повинен задовольнити нагальну потребу в створенні інтерактивних мережеских програм. У Java реалізовано кілька цікавих рішень, що дозволяють писати код, який виконує одночасно масу різних функцій і не забуває при цьому стежити за тим, що і коли має відбутися. У мові Java для вирішення проблеми синхронізації процесів застосований найбільш елегантний метод, який дозволяє конструювати

					КНТЕУ 121 06-24.БР	Лист
						9
Зм.	Лист	№ Докум.	Підпис	Дата		

прекрасні інтерактивні системи. Прості в обігу витончені підпроцеси Java дають можливість реалізувати в програмі конкретні перетворення, не відволікаючись при цьому на інтегрування глобальної циклічної обробки подій.

Творці Java наклали на мову і на середовище часу виконання кілька жорстких вимог, які дозволяють, одного разу написавши, завжди запустити програму в будь-якому місці і в будь-який час.

Мова Java, більш складна ніж мови командних інтерпретаторів але простіше для вивчення, ніж інші мови програмування, наприклад C ++.

Середовище Java - це щось набагато більше, ніж просто мова програмування. У неї вбудований набір ключових класів, що містять основні абстракції реального світу, з якими доведеться мати справу програмам. Основою популярності Java є вбудовані класи-абстракції, які зробили його мовою, дійсно незалежним від платформи. Бібліотеки прекрасно працюють на своїх платформах, однак сьогодні головною платформою стає мережа Інтернет.

Сьогодні все більше і більше розробників починають створювати корпоративні веб-додатки і використовувати переваги швидкості, захищеності і надійності, що забезпечуються серверними технологіями мови програмування Java.

1.3 Фреймворк Spring MVC для Java-платформи

Платформа Spring – популярна платформа додатків з відкритим кодом, призначена для спрощення розробки на J2EE. Вона складається з контейнера, платформи управління елементами і набору інтегрованих служб для веб-інтерфейсів користувача, транзакцій і збереження стану. До складу платформи Spring Web входить MVC – розширювана платформа для створення веб-додатків.

					КНТЕУ 121 06-24.БР	Лист
						10
Зм.	Лист	№ Докум._	Підпис	Дата		

Незважаючи на те, що Spring Framework не забезпечував якусь конкретну модель програмування, він є широко поширеним в Java-співтоваристві головним чином як альтернатива і заміна моделі Enterprise JavaBeans. Spring Framework надає велику свободу Java-розробникам в проектуванні, крім того, він надає добре документовані і легкі у використанні засоби вирішення проблем, що виникають при створенні додатків корпоративного масштабу.

Spring Framework забезпечує вирішення багатьох завдань, з якими стикаються Java-розробники і організації, які хочуть створити інформаційну систему, засновану на платформі Java. Через широкі функціональні можливості важко визначити найбільш значущі структурні елементи, з яких він складається. Spring Framework не повністю пов'язаний з платформою Java Enterprise, незважаючи на його масштабну інтеграцію з нею, що є важливою причиною його популярності. Spring Framework, ймовірно, найбільш відомий як джерело розширень (функцій), потрібних для ефективної розробки складних бізнес-додатків поза великовагових програмних моделей, які історично були домінуючими в промисловості. Ще однією перевагою є введення невикористовуваних функціональних можливостей в методах розробки, навіть поза платформи Java.

Цей фреймворк пропонує послідовну модель і робить її придатною до більшості типів додатків, які вже створені на основі платформи Java. Вважається, що Spring Framework реалізує модель Розробки, засновану на кращих стандартах індустрії, і робить її доступною в багатьох областях Java.

Основна перевага Spring – можливість розробки програми як набору легко зв'язаних компонентів. Чим менше компоненти програми знають один про одного, тим простіше розробляти новий і підтримувати існуючий функціонал програми. Класичний приклад - управління транзакціями. Spring дозволяє вам керувати транзакціями абсолютно незалежно від основної

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		11

логіки взаємодії з БД. Зміна цієї логіки не порушить транзакційність, так само як зміна логіки управління транзакціями не зламає логіку програми. Spring заохочує модульність. Компоненти можна додавати і видаляти незалежно один від одного. В принципі, додаток можна розробити таким чином, що він навіть не буде знати, що управляється за допомогою фреймворка Spring.

Принцип MVC у веб-програмуванні (Model - View - Controller, Модель – Вид – Контролер) – одна з найбільш вдалих ідей на сьогоднішній день. Принцип MVC інтуїтивно зрозумілий на перший погляд, але не дуже простий при поглибленні. Спочатку розглянемо, для чого він призначений. Принцип MVC, дозволяє розділити логіку реалізації, зовнішній вигляд (графічний інтерфейс, GUI) і взаємодія з користувачем. Це призводить до більш структурованого коду, дозволяє працювати над проектом більш спеціалізованим людям, спрощує підтримку коду, робить його більш логічним і зрозумілим. Зміна в одному з компонентів мінімально впливає на інші. Можна до однієї моделі підключати різні види, різні контролери. Розглянемо докладніше компоненти. Model (Модель) – містить логіку, обробку і верифікацію даних, звернення до баз даних. Модель не повинна безпосередньо взаємодіяти з користувачем. View (Вид) описує зовнішній вигляд програми. Controller (Контролер) - сполучна ланка між моделлю і видом, отримує дані від користувача, передає їх моделі, отримує оброблений результат і відображає його користувачу, зображено на (Рис. 1.1. Модель MVC):

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		12

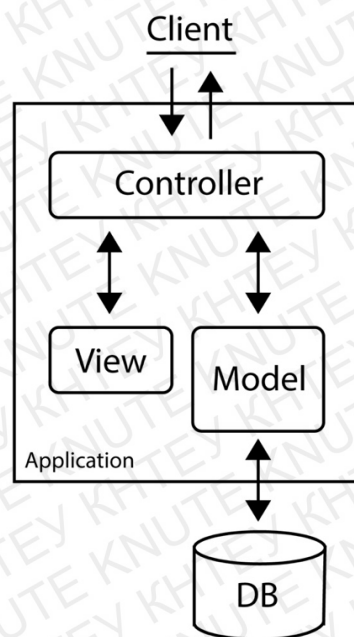


Рис. 1.1. Модель MVC.

Spring framework надає гарну можливість для реалізації архітектури MVC за допомогою Spring web MVC framework.

Spring web MVC framework забезпечує архітектуру модель - відображення - контролер (MVC) і готові компоненти, які можуть бути використані для розробки гнучких і слабо пов'язаних веб-додатків. Патерн MVC розділяє аспекти додатку (логіку введення, бізнес-логіку і логіку UI), забезпечуючи при цьому вільний зв'язок між ними:

1. Model (Модель) інкапсулює (об'єднує) дані програми, які будуть складатися з POJO (Plain Old Java Object).
2. View (Відображення, Вид) відповідає за відображення даних Моделі - генеруючи HTML, який відображається в браузері користувача.
3. Controller (Контролер) обробляє запит користувача, створює відповідну Модель і передає її для відображення в браузері користувача. Spring Web model-view-controller (MVC) framework розроблений навколо DispatcherServlet, який обробляє всі HTTP-запити і відповіді. Обробка запиту робочого процесу в Spring Web

Зм.	Лист	№ Докум.	Підпис	Дата

КНТЕУ 121 06-24.БР

Лист

13

4. MVC DispatcherServlet, зображено на (Рис. 1.2. Обробка запиту робочого процесу в Spring Web MVC DispatcherServlet) :

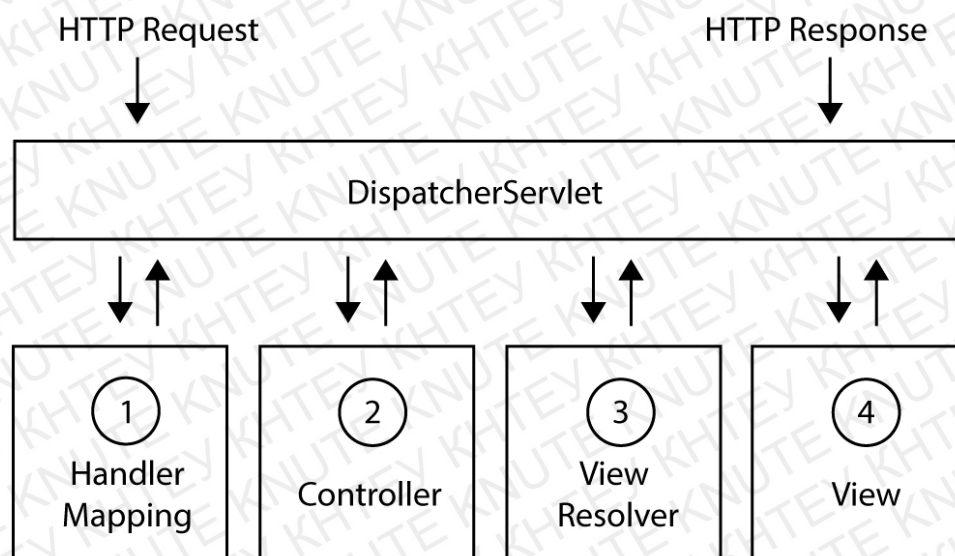


Рис. 1.2. Обробка запиту робочого процесу в Spring Web MVC DispatcherServlet

Нижче наводиться послідовність подій, відповідних вхідного запиту HTTP до DispatcherServlet:

1. Отримання запиту HTTP, DispatcherServlet консультиється з theHandlerMapping для виклику відповідного Контролера.
2. Контролер приймає запит і викликає відповідні методи сервісів на основі використовуваного методу GET або POST. Метод служби встановлює дані моделі, засновані на певній бізнес-логіці та повертає ім'я перегляду в DispatcherServlet.
3. За допомогою інтерфейсу ViewResolver DispatcherServlet визначає, який Вид потрібно використовувати на підставі отриманого імені.
4. Після того, як Вид (View) створений, DispatcherServlet відправляє дані Моделі у вигляді атрибутів в Вид, який в кінцевому підсумку відображається в браузері користувача.

Компоненти HandlerMapping, Controller и ViewResolver є частинами інтерфейсу WebApplicationContext і необхідні для створення веб-додатку.

1.4 Формування технічного завдання

Основною метою проекту є розробка програмного забезпечення мовою програмування Java за допомоги Spring Framework MVC. Додаток повинен працювати з будь-якого пристрою в найпопулярніших інтернет-браузерах. Мова інтерфейсу додатку: українська. Розроблений веб-додаток повинен дати змогу користувачам використовувати весь функціонал та фінансові транзакції. Вся особиста та фінансова інформація повинна зберігатися в базі даних. Дизайн додатку повинен бути виконаний з інтуїтивно зрозумілим інтерфейсом та в мінімалістичній стилістиці для максимізації комфорту користувача при використанні. Зовнішній вигляд веб-додатку повинен бути розроблений в трьох версіях: для користувачів мобільними пристроями, планшетами та комп'ютерами. Веб-додаток має складатися із таких сторінок:

1. Головна сторінка.
2. Сторінка балансу.
3. Сторінка історії транзакцій.
4. Сторінка налаштувань.

Головна сторінка має надавати користувачу змогу ознайомитися з продуктом, авторизуватися або зареєструватися.

Сторінка балансу має надавати можливості:

1. Перевірка фінансового балансу.
2. Створення нових платіжних рахунків для проведення анонімних фінансових транзакцій.
3. Історія останніх фінансових операцій.

Сторінка історії транзакцій матиме можливість відображення всієї фінансової історії транзакцій користувача на основі аналізу всіх створених

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		15

рахунків та успішних переказів.

Сторінка налаштувань дасть змогу користувачу створювати чи видаляти гаманці та групи гаманців.

1.5 Висновок до розділу 1

Кількість користувачів всесвітньої мережі Інтернет зростає по експоненті. Одна з найпоширеніших функцій мережі – оплата товарів та послуг онлайн. Для проведення оплати в мережі Інтернет найчастіше використовують online-платіжні системи. Найпопулярнішими стають платіжні рішення з мінімальними комісіями та швидкими фінансовими транзакціями. Тому було сформоване технічне завдання і вирішено реалізувати online-платіжну систему з миттєвими фінансовими транзакціями та відсутніми комісіями для виведення online-платіжної системи на новий рівень з метою приросту користувачів і вирішення конкретної потреби. Для охоплення більшої кількості потенціальних користувачів було вирішено використовувати крос-платформні рішення у розробці програмного забезпечення. Одним із найкращих рішень стало використання у розробці Spring MVC сумісно з об'єктно орієнтованою мовою програмування Java, що дозволяє розробити веб-додаток будь-якої складності та розширеним функціоналом.

					КНТЕУ 121 06-24.БР	Лист
						16
Зм.	Лист	№ Докум.	Підпис	Дата		

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Інтегроване середовище розробки програмного забезпечення для мови програмування Java

IntelliJ - одна з найпотужніших і популярних інтегрованих середовищ розробки (IDE) для мови програмування Java, що забезпечує швидку розробку і допомагає поліпшити якість коду.

IDE розшифровується як інтегроване середовище розробки. Це комбінація кількох інструментів, які роблять процес розробки програмного забезпечення більш простим, надійним і менш схильним до помилок. Має наступні переваги в порівнянні з текстовим редактором:

- 1) Інтеграція з корисними інструментами, такими як компілятор, відладчик, система контролю версій, інструменти збірки, різні платформи, профіліровщики додатків і так далі.
- 2) Підтримує функції навігації по коду, автозаповнення коду, рефакторінга і генерації коду, що прискорює процес розробки.
- 3) Підтримує модульне тестування, інтеграційне тестування і упрощення процесу написання коду за допомогою плагінів. Надає багатий набір плагінів для подальшого розширення функціональності IDE.

IntelliJ IDEA володіє найбільш ефективними функціями доповнення та корегування коду Java. Алгоритм прогнозування може точно передбачати зміст коду та в процесі написання доповнювати методом підстановки, що робить її таким унікальним середовищем розробки серед інших Java IDE:

- 1) Інтелектуальне завершення коду - підтримує контекстне завершення коду. Він дає список найбільш значущих символів, які можна

					<i>КНТЕУ 121 06-24.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		Криворучко О.В			Розробка online-платіжної системи	Стадія	Аркуш
Керівник		Цензура М.О.				Р2	17
Гарант		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група	
Розробив		Руденко А.О					
					Проектування програмного забезпечення		

застосувати в поточному контексті.

- 2) Ланцюгове завершення коду - це розширена функція завершення коду, яка перераховує відповідні символи, доступні через методи або методи отримання в поточному контексті.
- 3) Завершення статичного члена - дозволяє використовувати статичні методи або константи і автоматично додає необхідні оператори імпорту, щоб уникнути помилки компіляції.
- 4) Виявлення дублікатів - виявляє фрагменти дубльованого коду на льоту і повідомляє / пропозицію про це користувачеві.
- 5) Інспекції та швидкі виправлення. При виявленні, що можлива помилка в процесі написання коду, в одному рядку з'являється невелике лампове повідомлення, яка включає в собі список пропозицій.

Для організації робочого процесу IntelliJ IDEA пропонує набір інструментів, який включає в себе декомпілятор, засіб перегляду байт-коду, FTP і багато інших інструментів:

- 1) Контроль версій - IntelliJ підтримує більшість популярних систем контролю версій, таких як Git, Subversion, Mercurial, CVS, Perforce і TFS.
- 2) Інструменти збірки - IntelliJ підтримує Java та інструменти збірки, такі як Maven, Gradle, Ant, Gant, SBT, NPM, Webpack, Grunt і Gulp.
- 3) Тестовий прогін і відладка коду - IntelliJ IDEA дозволяє з легкістю виконувати модульне тестування. Середовище IDE включає в себе тестові прогони і інструменти відладки для основних середовищ тестування, включаючи JUnit, TestNG, Spock, Cucumber, ScalaTest, spec2 і Karma.
- 4) Декомпілятор - IntelliJ поставляється з вбудованим декомпілятором для класів Java, що дозволяє зазирнути всередину бібліотеки без

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		18

- 5) використання сторонніх плагінів.
- 6) Термінал - IntelliJ надає вбудований термінал.
- 7) Інструменти бази даних - IntelliJ надає інструменти бази даних, які дозволяють підключатися до діючих баз даних; виконувати запити; переглядати і оновлювати дані; і навіть управляти своїми схемами в візуальному інтерфейсі з самої IDE.
- 8) Сервер додатків - IntelliJ підтримує основні сервери додатків Tomcat, і багато інших. Ви можете розгорнути свої артефакти на серверах додатків і налагодити розгорнуті додатки в самій IDE.

Для розробки проекту було вибрано програмне забезпечення IntelliJ IDEA в результаті аналізу переваг, функціональних можливостей та комплексу інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, механізми інтеграції з середовищем тестування і системами управління версіями, унікальний інструмент оптимізації та перевірки коду, а також інноваційний візуальний конструктор графічних інтерфейсів.

2.2 Розробка архітектури та аналіз структури додатку

При створенні програмного забезпечення (ПЗ) одним із найважливіших етапів є розробка архітектури та аналіз структури додатку.

Архітектура додатку – це набір алгоритмів взаємодії компонентів структури системи програмного забезпечення. Архітектура включає:

- 1) Вибір структурних компонентів та їх інтерфейсів, за допомогою яких створена система.
- 2) З'єднання обраних компонентів структури і поведінки у все більш крупні системи.
- 3) Архітектурний стиль спрямовує всю організацію – елементів, їх

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		19

4) інтерфейсів та їх взаємодію.

Архітектура додатку відіграє велику роль на перших етапах. Додаток з правильною архітектурою простіше проаналізувати, редагувати, тестувати та масштабувати. Можна виділити список універсальних критеріїв для правильної архітектури: ефективність системи, гнучкість системи, можливість розширення системи, масштабування процесу розробки, тестування системи, повторне використання та простий супровід.

При розробці архітектури веб-додатку на стороні серверу (Рис. 2.2. Архітектура веб-додатку на стороні серверу) були використані:

- SPRING MVC – фреймворк для обробки HTTP запитів, що являється основною бізнес логікою UI архітектури.
- ORM – фреймворк, що дав змогу працювати з таблицями бази даних як з об'єктами.
- Spring DATA JPA – фреймворк використовувався для роботи з базою даних (відправка SQL запитів в базу даних).
- Spring Security – фреймворк для забезпечення безпеки веб-додатку.
- DB MySQL – база даних.

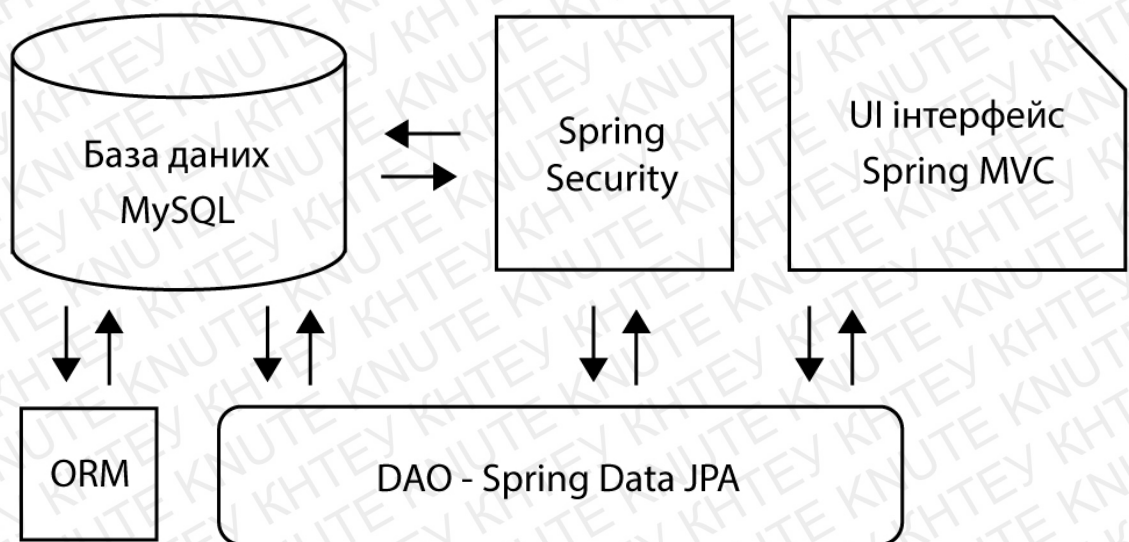


Рис. 2.2. Архітектура веб-додатку на стороні серверу.

При розробці архітектури веб-додатку на стороні клієнта використовувалася технологія JSP за допомогою якої на стороні сервера створювалися сторінки на мові веб-розмітки HTML до якого підключалися стилі CSS та динамічні скрипти мови JavaScript. Бібліотека jQuery використовувалася для спрощення створенню скриптів. Ajax використовувався для асинхронних запитів на сервер.

В процесі створення проекту було використана база даних MySQL 5 тому, що вона:

- Безкоштовна.
- Має велике ком'юніті.
- Гарно оптимізована.

Розробка баз даних починається з визначення таблиць та структури їх полів на основі яких будується логічна модель баз даних (Рис. 2.3. Логічна модель бази даних), логічна модель – це модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		21

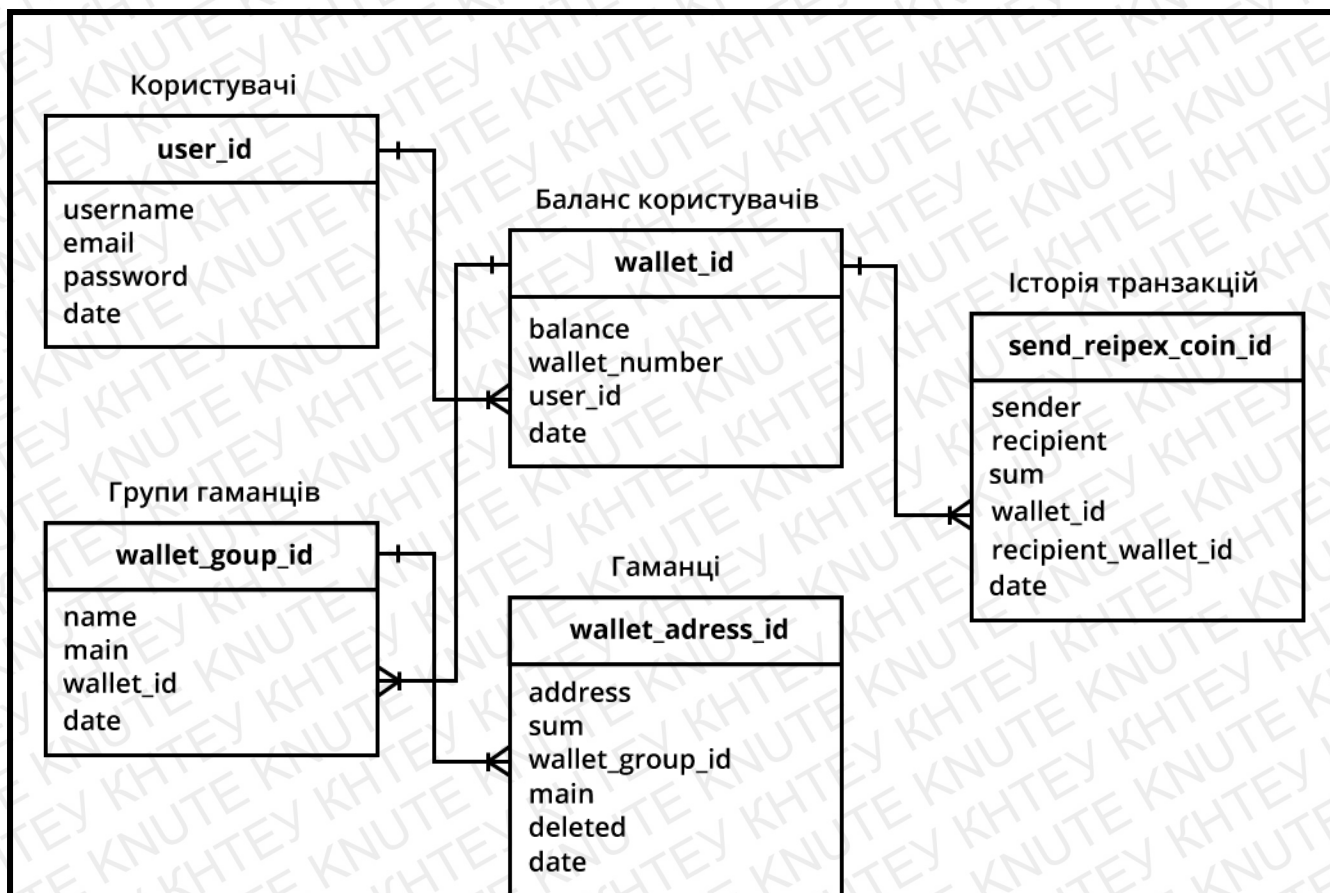


Рис. 2.3. Логічна модель бази даних

При створенні особистого кабінету користувача в першій таблиці бази даних «Користувачі» (Таблиця 2.1. Користувачі) створюється запис в полі `user_id` (нумерація) та `date` (дата реєстрації) автоматично, а поля `email` (електронна пошта) та `password` (пароль) записують дані введені користувачем під час реєстрації.

В другій таблиці «Баланс користувачів» (Таблиця 2.3. Баланс користувачів) створюється новий запис з полями `wallet_id` (ідентифікатор), `balance` (рахунок), `wallet_number` (унікальний адрес), `user_id` (прив'язка по зовнішньому ключу до таблиці `users` «Користувачі»), `date` (дата створення гаманця/папки адресів).

В наступній третій таблиці «Групи гаманців» (Таблиця 2.2. Групи гаманців) автоматично створюється новий запис з полями `wallet_group_id` (ідентифікатор), `name` (назва папки/гаманцю/групи), `main` (це поле відповідає

за визначення головна ця група чи ні (true или False)), wallet_id (поле за яким ця таблиця прив'язується до таблиці wallet_id «Баланс користувачів» по зовнішньому ключу), date (дата реєстрації цієї групи).

Створюється новий запис в четвертій таблиці «Гаманці» (Таблиця 2.4. Гаманці) з такими полями:

- wallet_address_id (ідентифікатор),
- address (адрес)
- sum (баланс адресу)
- wallet_group_id (прив'язка до таблиці wallet_group по зовнішньому ключу)
- main (цей поле відповідає за визначення головний це адрес чи ні (true или False))
- deleted (зрозуміти видалений адрес чи ні)
- date (дата створення).

В останню п'яту таблицю «Історія транзакцій» (Таблиця 2.5. Історія транзакцій) записується нові данні в поля sender (відправник), recipient (отримувач), sum (сума переказу), wallet_id (поле за яким ця таблиця прив'язується до таблиці «Баланс користувачів» по зовнішньому ключу), recipient_wallet_id (гаманець отримувача), date (дата створення транзакції).

Перед проведенням транзакції йде перевірка балансу в таблиці wallet_adress «Гаманці» в поле sum на наявність достатніх коштів, якщо коштів достатньо то проводиться транзакція і з адреси відправника списується сума з поля sum і нараховується в поле sum на адресу гаманця одержувача. Ця транзакція проводиться атомарно - якщо кошти списали і сталася помилка в базі даних або в результаті інших несправностей то стан таблиць повернеться до первісного вигляду.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		23

2.3 Проектування структури реляційної бази даних та її реалізація засобами MySQL

Реляційна база даних – база даних, заснована на реляційній моделі даних. В даний час ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні комерційні системи керування базами даних (СКБД).

Управління реляційними базами даних проекту було реалізоване завдяки **програмному забезпеченні MySQL**. База даних являє собою структуровану сукупність даних. Для запису, вибірки і обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базою даних, якою і є ПЗ MySQL. Оскільки комп'ютери справляються з обробкою великих обсягів даних, управління базами даних відіграє центральну роль в обчисленнях.

Для візуального проектування баз даних було використано **MySQL Workbench** – інструмент для інтегрування, проектування, моделювання, створення й експлуатації БД. Велика різноманітність реалізованих функцій **Java Spring** позбавляють від написання багаторядкового коду для користувача функцій на C або Pascal.

Створення інформаційної системи - це складний процес, в якому бере участь колектив розробників, які розбиваються на окремі робочі групи на стадії проектування для програмної реалізації.

У процесі створення інформаційної системи готуються робочі документи, що є основою для всіх розробників і користувачів системи.

Проектування бази даних полягає в багаторівневому описі майбутньої БД з різним ступенем деталізації і формалізації, в ході якого проводиться уточнення та оптимізація її структури.

Проектування включає опис предметної області і завдань інформаційної системи, далі йде логічний опис даних і побудова фізичної моделі БД. Розрізняють три етапи деталізації опису об'єктів БД і їх за

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		24

взаємозв'язків трьома основними рівнями моделювання системи - концептуальному, логічному і фізичному.

На концептуальному рівні проектування проводиться смислове (семантичне) опис інформаційного змісту предметної області, визначаються межі предметної області, проводиться абстрагування від несуттєвих для даної інформаційної системи деталей. В результаті визначаються як моделюються об'єкти, їх властивості та зв'язку. Виконується структуризація знань про предметну область, стандартизується термінологія.

На наступному кроці приймається рішення про те, в який СУБД буде реалізована БД. Визначальними параметрами є вид програмного продукту і категорія користувачів (професійні програмісти або кінцеві користувачі, або і те, і інше). Іншими показниками, що впливають на вибір СУБД, є: зручність і простота використання; якість засобів розробки, захисту і контролю БД; рівень комунікаційних засобів (застосування в мережах); фірма-розробник; вартість.

Кожна конкретна СУБД працює з певною моделлю даних. На логічному рівні проводиться відображення даних концептуальної моделі в логічну модель, підтримувану обраної СУБД. Тут об'єктом роботи виступають самі дані, їх структура і правила побудови. Логічна модель не залежить від конкретної СУБД, яка побудована на основі таблиць. Логічна модель може бути реалізована на будь-який СУБД реляційного типу.

На фізичному рівні проводиться вибір раціональної структури зберігання даних і методів доступу до них, вирішуються питання ефективного виконання запитів до БД, будуються додаткові структури, наприклад, індекси. У фізичній моделі міститься інформація про всі об'єкти БД (таблицях, індексах, процедурах і ін.) І використовуваних типах даних. Фізична модель залежить від конкретної СУБД. Однієї і тієї ж логічної моделі може відповідати кілька різних фізичних моделей (Рис. 2.4. Фізична

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		25

модель бази даних). Фізичне проектування є початковим етапом реалізації БД.

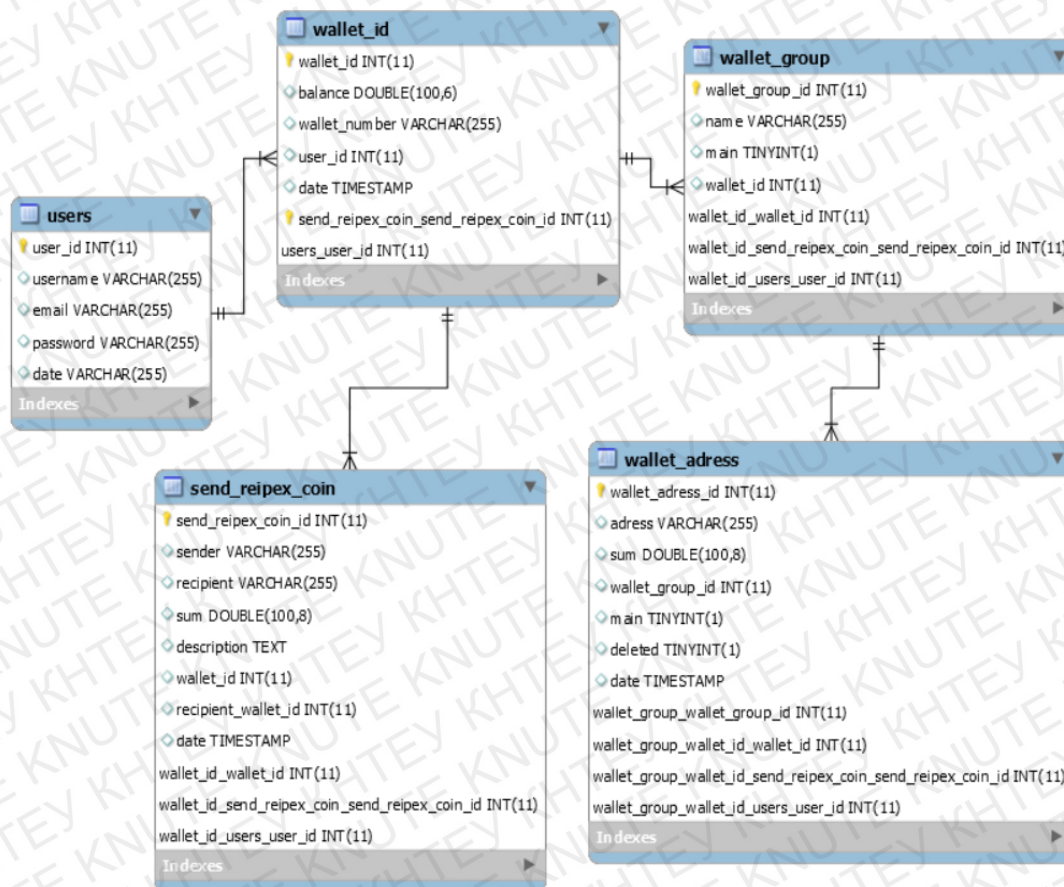


Рис. 2.4. Фізична модель бази даних

Структура і наповнення таблиць відображаються на Табл. 2.1. – Табл. 2.5.:

Таблиця 2.1.

КОРИСТУВАЧІ

user_id	username	firstname	secondname	referer	bounty_percent	email	phone	password	partner_program	email_verification	date
1	reipex			reipex	5	support@reipex.com		\$2a\$11\$Fk7CnVhRgFAdhrwVMWwV.z9MxKpk...	1	1	2017-11-30 18:51:33
1899	admin			reipex	5	andrey.rudenko@gmail.com		\$2a\$11\$TgpsiOqVkoV3vn9TYqmWge9goTl5Ch...	0	1	2020-04-21 21:45:34
1900	user			reipex	5	dzirunkaterina@gmail.com		\$2a\$11\$T4admkEGqIn36ggWkolBeoofVvfrPk...	0	1	2020-04-21 21:58:34
1901	diplom			reipex	5	diplom@gmail.com		\$2a\$11\$2.8JXGKg8BoIzKW79y23.RzCDdUHa...	0	1	2020-05-11 13:46:40

Таблиця 2.2

ГРУПИ ГАМАНЦІВ

wallet_group_id	name	main	wallet_id	date
1	Основной кошелёк	1	1	2017-11-30 18:51:33
27	Неосновной кошелек	0	1	2017-12-05 14:17:32
1930	Основной кошелек	1	1899	2020-05-17 20:46:58
1931	asdasd	0	1899	2020-04-21 21:48:43
1932	Основной кошелек	1	1900	2020-05-17 20:46:58
1933	Основной кошелек	1	1901	2020-05-11 14:06:44
1934	1	0	1901	2020-05-17 20:15:41

Таблиця 2.3.

БАЛАНС КОРИСТУВАЧІВ

wallet_id	balance	wallet_number	user_id	date
1	893900353.642100	0E890A-F31980E30D8EA813FA9925E-287A7C0...	1	2018-01-30 14:59:13
1899	149999.000000	D961C3D8-9E77B9-AB2EF-87D46954ED9-3939...	1899	2020-04-21 21:53:49
1900	1.000000	611457B266BC8A26F-B55A084960B8-BA07E-6...	1900	2020-04-21 21:58:34
1901	1000000.000000	15589FB63-1CED8FF9-8B082E560AD621-C454...	1901	2020-05-11 14:06:17

Таблиця 2.4.

ГАМАНЦІ

wallet_address_id	address	sum	wallet_group_id	main	deleted	date
1	lpgalxubbuikq3jg4lemlv45t63ka2y	893899353.64210010	1	1	0	2018-01-30 14:59:43
34	vs6nq5uim9xtdmzy3tbkowlc6f4vo0	1000.00000000	1	0	0	2017-12-05 14:18:04
40	inlsfq2wn9h1poqumd3s6nn4879jt0ij	0.00000000	1	0	0	2017-12-08 16:40:44
41	sagovqzly89d1unhegw518t2c5zkgva2	0.00000000	27	0	0	2017-12-08 16:40:52
134	h48osee67e46nx3uaektbdodqocc1enf	0.00000000	1	0	0	2017-12-24 10:54:29
140	8cqfm2quebta1wgjv73ifvmdm3n4z5r6	0.00000000	1	0	0	2017-12-26 08:58:51
141	4d41hlc3h5w0w8u6mubidwf9fjhp8vmf	0.00000000	1	0	0	2017-12-26 08:58:52
146	8p0xh64b0ib447wxu7gza7disd5aq1b	0.00000000	27	0	0	2017-12-28 11:13:53
147	0y1m0qtd7ffngclrxschlw5fnzyojhfo	0.00000000	1	0	0	2017-12-28 11:13:59
171	3zmw0047h19gbjq4s9rctgfrwwr53zk	0.00000000	1	0	0	2018-01-05 22:18:15
172	919f8uxvrvuiy9wbrvavduz9ovqje9	0.00000000	1	0	0	2018-01-05 22:53:59
261	pil7dso17yr2a6hdk38kgm8xffe075sm	0.00000000	1	0	0	2018-01-30 14:09:17
2007	4uffnkju1g3amfnlh0zehmz3zknrvio	0.00000000	1930	1	0	2020-04-21 21:45:34
2008	nzu65hqukvllh7bj9oc7vv6z3hxtemz2m	149999.00000000	1931	0	0	2020-04-21 21:55:40
2009	o7rm0j8moyxsfw7qbxj1a40fthbjdga	1.00000000	1932	1	0	2020-04-21 21:58:34
2010	fzk7v68ig75dylxknmk1u3apgf7hief	999999.00000000	1933	1	0	2020-05-17 20:23:35

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		27

Таблиця 2.5.

ІСТОРИЯ ТРАНЗАКЦІЙ

send_reipex_coin_id	sender	recipient	sum	description	bounty	wallet_id	recipient_wallet_id	date
1	lpgalxubbuikq3jg4emlv45t63ka2y	9v9ts6raj7wa1mdvue9h1tz3d7799ms7	150000.00000000		0	1	11	2017-12-01 17:53:02
2	lpgalxubbuikq3jg4emlv45t63ka2y	r7t4x88lzkq8mfxfhmx3j9lhosau0bs	5000.00000000		0	1	9	2017-12-01 17:54:37
3	lpgalxubbuikq3jg4emlv45t63ka2y	ypikqr29w9etbi30gfhqigghmk1p3xh	2000.00000000		0	1	3	2017-12-01 17:54:50
4	lpgalxubbuikq3jg4emlv45t63ka2y	mlasgd72a5xwdl933klwd5qzx35gwoal	2000.00000000		0	1	10	2017-12-01 17:55:23
5	lpgalxubbuikq3jg4emlv45t63ka2y	psh0xagv21m058nft7hchaxhutf5peyq	2000.00000000		0	1	4	2017-12-01 19:39:22
6	lpgalxubbuikq3jg4emlv45t63ka2y	r16y19g5d99fvuyekofx8lp4wcrpb0	10000.00000000		0	1	8	2017-12-02 09:44:25
7	lpgalxubbuikq3jg4emlv45t63ka2y	mlasgd72a5xwdl933klwd5qzx35gwoal	4000.00000000	Бонус +200%	0	1	10	2017-12-02 15:11:26
8	lpgalxubbuikq3jg4emlv45t63ka2y	ypikqr29w9etbi30gfhqigghmk1p3xh	4000.00000000	Бонус +200%	0	1	3	2017-12-02 15:13:11
9	lpgalxubbuikq3jg4emlv45t63ka2y	r7t4x88lzkq8mfxfhmx3j9lhosau0bs	10000.00000000	Бонус +200%	0	1	9	2017-12-02 15:14:34
10	lpgalxubbuikq3jg4emlv45t63ka2y	psh0xagv21m058nft7hchaxhutf5peyq	4000.00000000	Бонус +200%	0	1	4	2017-12-02 15:16:10
11	lpgalxubbuikq3jg4emlv45t63ka2y	ydeow72ju2o602kdcqqr4d8vj386ids	500.00000000		0	1	12	2017-12-02 18:45:45
12	lpgalxubbuikq3jg4emlv45t63ka2y	quvcgr0rw8kintex6tyl5qs7kk6td9lu	500.00000000		0	1	19	2017-12-03 15:06:04
13	lpgalxubbuikq3jg4emlv45t63ka2y	cp0b92zfs9a4j63qc3u7aeodp0pm33qr	1850.00000000		0	1	18	2017-12-03 17:59:38
14	lpgalxubbuikq3jg4emlv45t63ka2y	cp0b92zfs9a4j63qc3u7aeodp0pm33qr	3700.00000000	Бонус +200%	0	1	18	2017-12-03 17:59:38
15	lpgalxubbuikq3jg4emlv45t63ka2y	am98qrvmahhlzqjr4nnic7nw00xerx	1100.00000000		0	1	22	2017-12-04 18:23:35
16	lpgalxubbuikq3jg4emlv45t63ka2y	am98qrvmahhlzqjr4nnic7nw00xerx	2200.00000000	Бонус +200%	0	1	22	2017-12-04 18:23:36

2.4 Моделювання графічного інтерфейсу додатку

При виборі програмного забезпечення все частіше користувачі звертають увагу на легкість в використанні та дизайн додатку, що грає велику роль у майбутній популярності додатку та прирості користувачів. В майбутньому користувачі можуть монетизувати затратений фінансовий бюджет на маркетинг та дати змогу розвиватися значніше швидше додатку без потреби пошуку інвесторів чи отримувати кредитування на розвиток. Дизайн додатку та легкість в використанні доволі важлива тенденція сьогоденних реалій. Користувач звертає увагу в першу чергу на обкладинку, котру розробники з легкістю можуть доповнити функціоналом. Найважливішим є визвати у користувача велику кількість позитивних емоцій за перші 3 секунди, дати бачення як він зможе вирішити свої проблеми залишаючись та використовуючи веб-додаток. Це не означає, що додаток повинен мати тільки гарний дизайн, але стартапи і компанії котрі нехтують зовнішнім видом додатка, розробкою брендбука та елементами фірмового стилю - втрачають доволі великий сегмент аудиторії, котра все більш стає вимогливою і обізнаною в трендах. Для створення якісного інтерфейсу користувача у команді розробників обов'язково повинен бути -

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		28

залучений дизайнер проектувальник. Такий професіонал здатний вирішити багато проблем так як саме він стандартизує зовнішній вигляд додатку: кнопки, іконки, оформлення тексту та інші елементи інтерфейсу, створюючи єдину екосистему в якій користувачу буде приємно проводити час використовуючи функціонал додатку.

В процесі проектування графічного інтерфейсу веб-додатку були використані темні кольори та мінімалістичний стиль, що відповідає тренду і та дозволяє знайде свою аудиторію користувачів. Макети інтерфейсу були розроблені завдяки графічному редактору Adobe Illustration – лідер ринку в галузі комерційних засобів редагування векторних зображень. Макети дозволяють заздалегідь продумати архітектуру графічного інтерфейсу.

Головний функціонал веб-додатку складається із шести основних сторінок:

- 1) Сторінка авторизації.
- 2) Сторінка реєстрації.
- 3) Головна сторінка платіжної системи.
- 4) Сторінка створення транзакцій.
- 5) Сторінка історії платіжних транзакцій.
- 6) Сторінка налаштувань платіжної системи.

На сторінці авторизації розташовані поля вводу особистих даних «Email» та «Пароль», та кнопка «Ввійти» для входу в особистий кабінет користувача (Рис. 2.5. Авторизація в платіжній системі).

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		29

Рис. 2.5. Авторизація в платіжній системі

На сторінці реєстрації розташовані поля вводу «Ім'я користувача», «Email», «Пароль», «Повторити пароль» та кнопка «Зареєструватися» для створення особистого кабінету користувача і подальшого входу у платіжну систему (Рис. 2.6. Реєстрація в платіжній системі).

Рис. 2.6. Реєстрація в платіжній системі

					КНТЕУ 121 06-24.БР		Лист
							30
Зм.	Лист	№ Докум._	Підпис	Дата			

Головна сторінка платіжної системи складається з меню, назви веб-додатку, балансу, гаманців (папки адресів), рахунків (адресів) (Рис. 2.7. Головна сторінка особистого кабінету користувача платіжної системи).

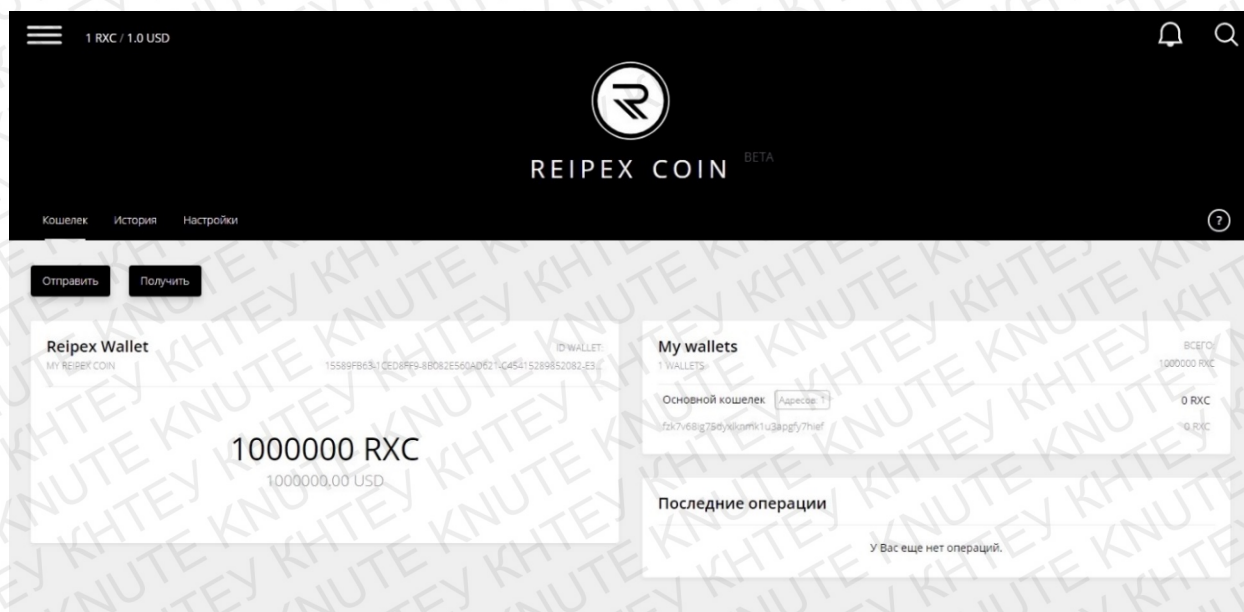


Рис. 2.7. Головна сторінка особистого кабінету користувача платіжної системи

Сторінка створення транзакцій включає в себе поля вводу «З якого гаманця», «Кому», «Сума» та «Опис» (Рис. 2.8. Створення платіжної транзакції).

					КНТЕУ 121 06-24.БР	Лист
						31
Зм.	Лист	№ Докум.	Підпис	Дата		

REIPEX COIN

От:

Кому:

Сумма: RXC

Описание:

Далее

Рис. 2.8. Створення платіжної транзакції

Сторінка підтвердження транзакцій відображає інформацію для підтвердження введених даних користувачем з метою створення транзакції (Рис. 2.9. Підтвердження платіжної транзакції).

Подтверждение

Сумма:
1 RXC / 1USD

Отправитель:
fzk7v68ig75dyxlknmk1u3apgy7hief

Получатель:
7okc8tno915gtm96yhvu65mlyo6zqztu

Подтвердить

Назад

Рис. 2.9. Підтвердження платіжної транзакції

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		32

Сторінка історії платіжних транзакцій складається із таблиці в яку входять наступні стовпці даних «Тип операції», «Дата», «Сума», «Кому/Від кого», «Опис», (Рис. 2.10. Історія платіжних транзакцій).

1 RXC / 1.0 USD

REIPEX COIN

BETA

Кошелек

История

Настройки

Отправить

Получить

История

Все

Отправленные

Полученные

Тип операции	Дата	Сумма	Кому / От	Описание
Отправлено	17/05/2020 20:32	-1.0 RXC	70kc8tno915gtm96yhwu65mlyo6zqtu / fzK7v68lg75dylknmk1u3apgy7hief	Выполнено
Получено	17/05/2020 20:32	+1.0 RXC	70kc8tno915gtm96yhwu65mlyo6zqtu / fzK7v68lg75dylknmk1u3apgy7hief	Выполнено

Рис. 2.10. Історія платіжних транзакцій

Сторінка налаштувань платіжної системи включає в себе можливість створювати та видаляти папки гаманців та гаманці (Рис. 2.11. Створення гаманців платіжної системи) і (Рис. 2.12. Видалення гаманців платіжної системи).

My wallets		
Название	Сумма	
Основной кошелек	999999 RXC	
1	1 RXC	Добавить адрес

ID кошелька

15589f9b63-1c5d4ff9-8d982e5604d621c45415289652082-6354EE165DCEB9

ID кошелька - это ваш уникальный идентификатор. Он полностью индивидуален, используется для доступа к личному кошельку.

Рис. 2.11. Створення гаманців платіжної системи

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		33

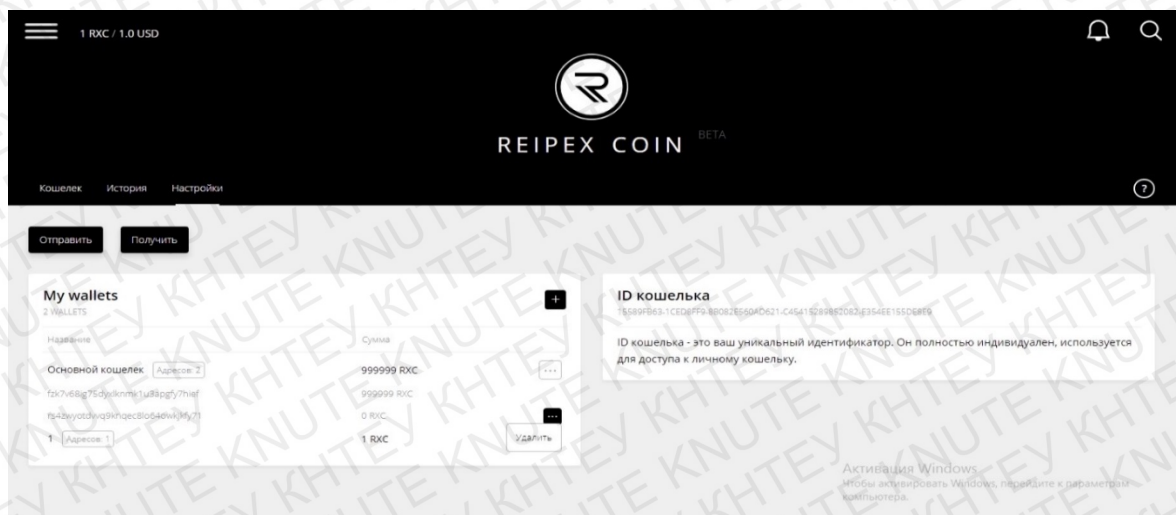


Рис. 2.12. Видалення гарантів платіжної системи

Процес моделювання графічного інтерфейсу допоміг розробити концепцію реалізації майбутнього програмного забезпечення метою якого було об'єднання сучасного дизайну та функціональності веб-додатку для користувачів.

2.5 Висновки до розділу 2

Для розробки веб-додатку на мові програмування Java було вибрано середовище розробки програмного забезпечення IntelliJ. Спроектована структура реляційної бази даних та реалізована засобами MySQL. Найважливіший та найскладніший етап в проекті – це формування технічного завдання та розробка архітектури веб-додатку, побудова логічної та фізичної моделі бази даних, що мають великий вплив на складність подальшої розробки програмного забезпечення. При виконанні цього етапу потрібно було мати розуміння як вирішувати поставлену мету та баченням фінального продукту.

Саме після проектування архітектури веб-додатку був створений та підготовлений до активної стадії розробки проект, що дозволило розробити графічний інтерфейс та бізнес логіку веб-додатку засобами середовища Spring MVC, Spring Data, Spring Security.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		34

РОЗДІЛ 3.

ПРОГРАМНА РЕАЛІЗАЦІЯ ON-LINE ПЛАТІЖНОЇ СИСТЕМИ

3.1 Графічний інтерфейс розробленого додатку

Програмна реалізація веб-додатку складається з декількох слоїв, **Controler, Sevice** та **DAO**:

Controler (Spring MVC) – отримання та обробка запитів.

Service (JAVA) – бізнес-логіка.

DAO (Spring DATA) – робота з базою даних.

Веб-додаток створений за допомогою такої структури:

- 1) **UserControler/Sevice/DAO.java** використовується для роботи з користувачами: реєстрація та авторизація.
- 2) **WalletController/Sevice/DAO.java** використовується для роботи з гаманцями: додавання та видалення.
- 3) **ReipexCoinOperationsController/Sevice/DAO.java** використовується для створення транзакцій.
- 4) **NavigationController.java** використовується для навігації по веб-додатку.

Для відображення інтерфейсу користувача створено файли із форматом .jsp (сукупність html та java операторів), котрі знаходяться в пакеті «WebApp» (Рис. 3.1. Код інтерфейсу користувача).

					КНТЕУ 121 06-24.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка online-платіжної системи	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				РЗ	35	47
Керівник		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розробив		Руденко А.О			Програмна реалізація on-line платіжної системи			

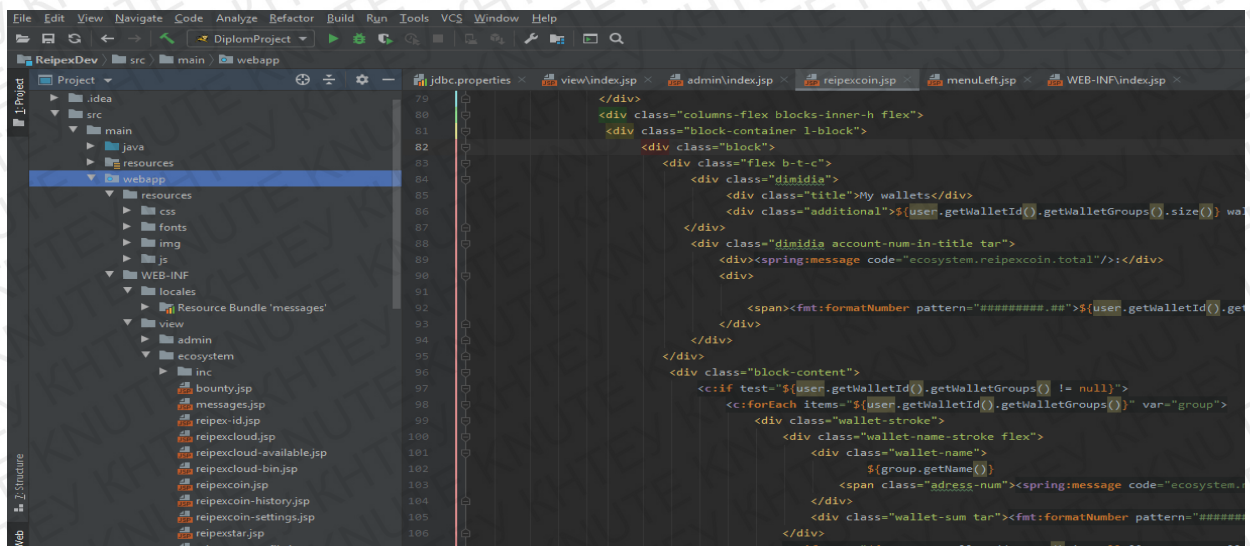


Рис. 3.1. Код інтерфейсу користувача

Сторінка «Авторизація» та «Реєстрація» дозволяє користувачу створити особистий обліковий запис в платіжній системі завдяки функції Реєстрація або увійти в особистий кабінет за допомогою функції Авторизації.

Сторінка «Реєстрація» (Рис. 2.6. Реєстрація в платіжній системі) **registrationform.jsp** складається з полів для вводу особистих даних «Ім'я користувача», «Email», «Пароль» та «Підтвердження паролю», котрі передаються на сервер по протоколу **http** методом **post** (Додаток А. Реєстрація) та перевіряються за допомогою валідатора перед збереженням в базу даних.

Сторінка «Авторизації» (Рис. 2.5. Авторизація в платіжній системі) **loginform.jsp** складається з полів для вводу «Email» та «Пароль», які перевіряються на коректність та збіг введених даних на сервері в базі даних методом **post** (Додаток Б. Авторизація). Якщо дані співпадають то створюється сесія користувача завдяки фреймворку **Spring Security**, інакше виводиться модальне вікно з повідомленням про помилку у введених даних. Код якого знаходиться в файлі **modalWrongPassword.jsp**.

При виході користувача с особистого кабінету (закінченні сесії) куки сесії видаляються за допомогою фреймворка **Spring Security** та надстройки

									Лист
									36
Зм.	Лист	№ Докум.	Підпис	Дата					

<logout logout-success-url="/index?logout" delete-cookies="JSESSIONID"/>.

В результаті створення особистого кабінету за допомогою фреймворка **Spring Security** для користувача автоматично створюється та присвоюється унікальний ідентифікатор гаманця **wallet_number** (Додаток В. Ідентифікатор) в таблиці бази даних **wallet_id** (Таблиця 2.3. Баланс користувачів) за допомогою коду **WalletId walletId = walletService.generateWalletId(user)**. Автоматично створюється група гаманців **wallet_group** з присвоєнням назви «Основний гаманець» за допомогою коду **walletService.addWalletGroup ("Основний гаманець", walletId, true)**; та основного рахунку **address**, який генерується (Додаток Г. Створення основного рахунку) і зберігається в таблицю бази даних **wallet_address** (Таблиця 2.4. Гаманці) за допомогою коду **WalletAddress walletAddress = walletService.addWalletAddress(walletGroup, true)**, що дозволяє користувачу зразу після Реєстрації використовувати веб-додаток в повному обсязі.

Основною функцією веб-додатку є можливість створювати транзакції між клієнтами програмного продукту при натисканні на кнопку «Відправити» створюється об'єкт **SendReipexCoin** транзакції з введеними даними в поля вводу користувачем та передача в функцію **sendReipexCoinService**.

makeTransaction(sendReipexCoin) (Додаток Д. Створення транзакції).

Для створення транзакції користувач вводить всі данні в поля вводу (Рис. 2.7. Створення платіжної транзакції) та натискає на кнопку «Далі», що відображає всі введенні дані для перевірки коректності даних (Рис. 2.9. Підтвердження платіжної транзакції). Всі введенні користувачем дані звіряються з базою даних для перевірки коректності введених, а також відбувається процес перевірки балансу на наявність коштів для переказу.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум._	Підпис	Дата		37

В результаті успішної транзакції на баланс отримувача додаються кошти (Додаток Е. Зачислення коштів), а з балансу відправника сума транзакції віднімається (Додаток Є. Списання коштів).

Сторінка «Історія транзакцій» (Рис. 2.10. Історія платіжних транзакцій) дозволяє користувачу аналізувати свої перекази завдяки виведенню збережених даних з MySQL за допомогою коду **List<SendReipexCoin> allHistory = sendReipexCoinService.getAllHistory(user.getWalletId());**

Додавання гаманців та груп гаманців можлива на сторінці «Налаштування» при натисканні користувачем кнопки «Додати» (Рис. 2.11. – 2.12.) завдяки коду **wallet Service. Add Wallet Address (walletService.addWallet Group (name, user.get WalletId()))**. Видалення гаманця чи груп гаманців реалізується при натисканні кнопки «Видалити» завдяки розробленому коду **walletService.deleteWalletAddressById(Long.parseLong(id),user.getWalletId ())** (Додаток Ж. Видалення чи додавання гаманців).

3.2 Архітектура розробленого додатку та інструкція користувача

Архітектура веб-додатку розроблена таким чином, щоб користувач інтуїтивно і з мінімальною кількістю часу та дій отримував бажаний результат. В першу чергу для створення особистого кабінету користувачу потрібно перейти на сторінку Реєстрації (Рис. 2.6. Реєстрація в платіжній системі) та заповнити всі поля особистих даних «Ім'я користувача», «Email», «Пароль» та «Підтвердження паролю». Наступним етапом після створення особистого кабінету є Авторизація (Рис. 2.5. Авторизація в платіжній системі) перейшовши на сторінку та заповнити обов'язкові поля даними введених при реєстрації: «Email» та «Пароль».

Після Авторизації користувач потрапляє в особистий кабінет (Рис. 2.7. Головна сторінка особистого кабінету користувача платіжної системи) де відображаються основні дані веб-додатку: «Баланс», «Гаманці та групи

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		38

гаманців» та «Останні операції». При натисканні на кнопку «Відправити», відкривається діалогове вікно з полями для вводу (Рис. 2.8. Створення платіжної транзакції): «З якого гаманця», «Кому», «Сума», «Опис» та кнопка «Далі». В результаті натискання на кнопку «Далі» користувач переходить на наступне діалогове вікно для перевірки введених даних (Рис. 2.9. Підтвердження платіжної транзакції).

При переході на сторінку Історія транзакцій (Рис. 2.10. Історія платіжних транзакцій) користувач має можливість проаналізувати свої перекази завдяки виведенню наступних даних: «Тип операції», «Дата», «Сума», «Кому/Від кого» та «Опис».

Сторінка налаштувань (Рис. 2.11. Створення гаманців платіжної системи) та (Рис. 2.12. Видалення гаманців платіжної системи) надає змогу користувачу створювати гаманці та групи гаманців чи видаляти їх завдяки натисканню на кнопки «Додати» чи «Видалити».

Для виходу користувача з особистого кабінету потрібно натиснути на бокове меню з ліва та натиснути на кнопку «Вихід».

3.3. Висновки до розділу 3

У даному розділі розглянута програмна реалізація веб-додатку за допомогою мови програмування **Java** та фреймворків **Spring MVC**, **Spring Data** і **Spring Security**, його структура (**Controler**, **Service**, **DAO**) та функціональні можливості користувачів при взаємодії з системою. Створена інструкція користувача для повного розуміння веб-додатку та можливостей в використанні. Основна частина розробки програмного продукту відобразила весь потенціал та бачення майбутнього розвитку.

Розроблена online-платіжна система при використанні сучасних інструментів розробки з урахуванням спрощеної логіки інтерфейсу для мінімізації часу користувача при продуктивній взаємодії з веб-додатком.

					КНТЕУ 121 06-24.БР	Лист
Зм.	Лист	№ Докум.	Підпис	Дата		39

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання дипломного проекту було створено програмний продукт на основі використання сучасних технологій та аналізу потреб ринку. Вибір об'єктно-орієнтованої мови програмування Java та Spring Framework дозволив розробити версію веб-додатку для користувачів мережі Інтернет орієнтованих на використання мобільних пристроїв та стільникових смартфонів.

Представлені основи розробки платіжної системи, об'єктно-орієнтованої мови програмування Java та платформи Spring Framework. Для розробки веб-додатку на мові програмування Java було вибране середовище IntelliJ. Спроектована структура реляційної бази даних та реалізована засобами MySQL. Сформоване технічне завдання та розроблена архітектура веб-додатку, побудована логічна та фізична модель бази даних. Розроблено графічний інтерфейс та бізнес логіку веб-додатку засобами середовища Spring IoC, Spring MVC, Spring Data, Spring Security.

Розглянута програмна реалізація веб-додатку за допомогою мови програмування Java та фреймворків, його структура та функціональні можливості користувачів при взаємодії з системою. Створена інструкція користувача для розуміння роботи веб-додатку та можливостей його використання. Розроблена online-платіжна система при використанні надійних та сучасних інструментів розробки.

Перспективою розробленої архітектури є можливість розробляти слабозв'язанні і легкозмінні модулі програми за допомогою патерну MVC та dependency Injection (IoC).

					КНТЕУ 121 06-24.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка online-платіжної системи	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				ВП	40	47
Керівник		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розробив		Руденко А.О			Висновки та пропозиції			

Перенести розроблений додаток на мікросервісну архітектуру для зниження навантаження на сервер з метою підвищення продуктивності при високих навантаженнях на веб-додаток.

Весь графічний інтерфейс (UI) буде знаходитися на клієнті і відправляти на сервер REST: запити і отримувати відповіді по HTTP протоколу.

Також завдяки розробленому адаптивному інтерфейсу та архітектурі орієнтованій на користувача, веб-додаток можливо розробити для мобільних та десктопних пристроїв, що дозволить збільшити кількість цільової аудиторії.

					КНТЕУ 121 06-24.БР	Лист
						41
Зм.	Лист	№ Докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Васильєв О. Програмування мовою Java. / О. Васильєв - В.: «Навчальна книга Богдан», 2020. - 696 ст.
2. Грофф Д. SQL. Повне керування. / Д. Грофф, П. Вайнберг, Е. Дж. Оппель. - В.: «Вільямс», 2015. - 960 ст.
3. Гослінг Д. Язык програмування Java SE 8. / Д. Гослінг, Б. Джой, Г. Стіл, Г. Брача, А. Баклі - В.: «Вільямс», 2015. – 672 с.
4. Шілдрт Г. Java 8. Повне керування. / Г. Шілдрт - В.: «Вільямс», 2017. - 1376 ст.
5. Шефер К. Spring 5 для професіоналов. / К. Шефер, К. Хо, Р. Харроп – В.: «Вільямс», 2016. – 1120 ст.

Інтернет ресурси:

1. Spring Famework Documentation, [Електронний ресурс]. – Режим доступу: <http://docs.spring.io/>
2. IntelliJ IDEA, [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/>
3. Електронний посібник з мови розмітки гіпертексту HTML [Електронний ресурс]. Режим доступу: <http://htmlbook.ru/html5>
4. Електронний посібник по CSS [Електронний ресурс]. Режим доступу: <http://www.w3schools.com/css/>
5. Електронний посібник JavaScript [Електронний ресурс]. Режим доступу: <https://learn.javascript.ru/>

					<i>КНТЕУ 121 06-24.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		Криворучко О.В			Розробка online-платіжної системи	Стадія	Аркуш
Керівник		Цензура М.О.				СД	42
Гарант		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група	
Розробив		Руденко А.О					
					Список використаних джерел		

ДОДАТКИ

Додаток А

ЛІСТИНГ КОДУ СТОРІНКИ «РЕЄСТРАЦІЯ»

```
<form action="registration" method="post" id="registration-form">
<input type="hidden" name="{_csrf.parameterName}" value="{_csrf.token}"
/>
<input type="text" id="username" onchange="checkField(this)" class="common-
modal-input validation" placeholder="<spring:message
code='registrationform.name'/>" required name="username">
<input type="text" id="email" onchange="checkField(this)" class="common-
modal-input validation" placeholder="E-mail" required name="email">
<input type="password" id="password" onchange="checkField(this)"
class="common-modal-input validation" placeholder="<spring:message
code='registrationform.pas'/>" required name="password">
<input type="password" id="confirmPassword" onchange="checkField(this)"
class="common-modal-input validation" placeholder="<spring:message
code='registrationform.pas2'/>" required name="confirmPassword">
<button class="btn btn-grey-b" onclick="checkFields()"
type="button"><spring:message code='registrationform.btn'/></button>
</form>
```

ЛІСТИНГ КОДУ СТОРІНКИ «АВТОРИЗАЦІЯ»

```
<form action="login" method="post" id="login">
<input type="hidden" name="{_csrf.parameterName}" value="{_csrf.token}"
/>
<input type="email" class="common-modal-input" name="username"
placeholder="E-mail" required>
<input type="password" class="common-modal-input" name="password"
placeholder="<spring:message code='loginform.pas1'/>" required>
<button class="btn btn-grey-b" type="submit"><spring:message
code="loginform.log"/></button>
</form>
```

ЛІСТИНГ КОДУ СТВОРЕННЯ ІДЕНТИФІКАТОРА ГАМАНЦЯ

```
public String generateId(String phrase) {  
    String[] words = phrase.split("\\s+");  
    StringBuilder generatedId = new StringBuilder(64);  
    for (String word : words) {  
        generatedId.append(sha256(word, true));  
    }  
    generatedId = new  
    StringBuilder(sha256(base64(generatedId.append(System.currentTimeMillis()).toS  
tring()).toUpperCase()));  
    int localCounter = 0;  
    for (int i = 0; i < 64; i++) {  
        if (localCounter > 5 && i != 63 && tripleBooleanCheck()) {  
            generatedId.setCharAt(i, '-');  
            localCounter = 0;  
        }  
        localCounter++;  
    }  
    return generatedId.toString();  
}
```

ЛІСТИНГ КОДУ СТВОРЕННЯ ОСНОВНОГО РАХУНКУ

```
public static String generateAddress() {  
    String address = "000";  
    StringBuilder temp = new StringBuilder();  
    Random random = new Random();  
    while (!address.startsWith("1f")) {  
        for (int i = 0; i < 128; i++) {  
            char c = chars[random.nextInt(chars.length)];  
            temp.append(c);  
            // address = HashingUtil.getHash(temp.toString());  
            address = temp.toString();  
        }  
        address = address.substring(0, 32);  
        return address;  
    }  
    return address;  
}
```

ЛІСТИНГ КОДУ СТВОРЕННЯ ТРАНЗАКЦІЇ

```
public boolean makeTransaction(SendReipexCoin sendReipexCoin) {  
    double sum = sendReipexCoin.getSum();  
    String recipient = sendReipexCoin.getRecipient();  
    String sender = sendReipexCoin.getSender();  
    WalletAddress walletAddressRecipient =  
        walletService.findWalletAddressByAddress(recipient);  
    if(recipient != null && walletAddressRecipient != null){  
        walletService.subCoin(sender, sum);  
        walletService.addCoin(recipient, sum);  
        sendReipexCoinDAO.save(sendReipexCoin);  
        return true;  
    }  
    return false;  
}
```

ЛІСТИНГ КОДУ ЗАЧИСЛЕННЯ КОШТІВ

```
public boolean addCoin(String address, double sum){  
    WalletAddress walletAddress = walletAddressDAO.findByAddress(address);  
    if(walletAddress != null){  
        walletAddress.setSum(walletAddress.getSum() + sum);  
        save(walletAddress);  
        WalletId walletId = walletAddress.getWalletGroup().getWalletId();  
        walletId.setBalance(walletId.getBalance() + sum);  
        save(walletId);  
        return true;  
    }  
    return false;  
}
```

ЛІСТИНГ КОДУ СПИСАННЯ КОШТІВ

```
public boolean subCoin(String address, double sum) {  
    WalletAddress walletAddress = walletAddressDAO.findByAddress(address);  
    if (walletAddress != null){  
        if(walletAddress.getSum() >= sum){  
            walletAddress.setSum(walletAddress.getSum() - sum);  
            save(walletAddress);  
            WalletId walletId = walletAddress.getWalletGroup().getWalletId();  
            walletId.setBalance(walletId.getBalance() - sum);  
            save(walletId);  
            return true;  
        }  
    }  
    return false;  
}
```

ЛІСТИНГ КОДУ ВИДАЛЕННЯ ЧИ ДОДАВАННЯ ГАМАНЦІВ

```
public boolean deleteWalletAddressById(Long id, WalletId walletId) {  
    WalletAddress walletAddress = findWalletAddressById(id);  
    if(walletAddress != null &&  
        walletAddress.getWalletGroup().getWalletId().getWalletNumber().equals(walletId  
            .getWalletNumber())){  
        deleteWalletAddressById(id);  
        return true;  
    }  
    return false;  
}
```