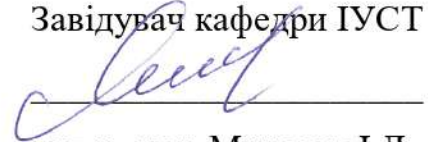


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелев І.Л.

" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

Розроблення автоматизованого робочого місця інвестора ІТ-проєкту

Виконав: студент групи 4141


(підпис)

Глаголев Д.Р.

Керівник роботи:

Викладач кафедри ІУСТ

(посада, науковий ступінь вчене звання)



Бобровський О.С.

(підпис)

м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми



к.т.н, доц. Гайда А.Ю.

"___" _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "бакалавр"

Студенту _____ Глаголеву Данилу Ростиславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення автоматизованого робочого місця інвестора ІТ-проекту

Керівник роботи Викладач кафедри ІУСТ Боброський Олексій Сергійович

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 "червня" 2025 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали переддипломної практики

4. Перелік питань, що належать до розробки (найменування розділів) Вступ, 1. Розробка концепції автоматизованого робочого місця інвестора ІТ-проекту, 1.1. Опис об'єкта дослідження та аналіз предметної області, 1.2. Формування вимог до АРМ інвестора ІТ-проекту, 1.3 Розробка концепції АРМ інвестора, 2. Розробка проєкта інформаційної системи, 2.1. Загальносистемні рішення, 2.2. Рішення з інформаційного забезпечення, 2.3. Рішення з математичного забезпечення, 2.4. Рішення з програмного забезпечення, 2.5. Рішення з технічного забезпечення, 3. Реалізація проєкту інформаційної системи, 3.1. Розробка робочої документації по експлуатації системи, 3.2. Екранні форми, 4. Охорона праці, Висновки, Список використаних джерел, Додатки.

5. Перелік презентаційних матеріалів 2. Актуальність, об'єкт та предмет дослідження, 3. Мета та завдання, 4. Модель концепції автоматизованого робочого місця інвестора ІТ-проекту, 5. Концептуальна модель, 6. Структура бази даних, 7. Математична модель, 8. Діаграма компонентів, 9.-10. Ключові пункти етапу розробки робочої документації, 11. Особливості створення екранних форм інтерфейсу користувача, 12. Покрокове виконання програми "Автоматизоване робоче місце інвестора ІТ-проекту", 13.-14. Екранні форми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Бобровський О.С.	03.02.2025	22.02.2025
2	Бобровський О.С.	22.02.2025	21.03.2025
3	Бобровський О.С.	21.03.2025	25.04.2025
4	Бобровський О.С.	25.04.2025	07.06.2025

7. Дата видачі завдання 03 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	13.02.2025	
2	Постанова задачі	15.02.2025	
3	Розробка концепції	22.02.2025	
4	Розробка проекту ІС	21.03.2025	
5	Реалізація проекту	25.04.2025	
6	Розробка питань з охорони праці	07.06.2025	
7	Розробка графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

Студент



(підпис)

Глаголев Д.Р.

(ПІБ)

Керівник роботи



(підпис)

Бобровський О.С.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці програмної системи для управління проектами, яка буде орієнтована на інвесторів та керівників бізнесу. У першому розділі проведено аналіз предметної області, за результатами аналізу розроблено концепцію програмної системи та сформульовано вимоги, постанову задачі та технічне завдання. У другому розділі розроблено проєктні рішення з інформаційного, програмного та технічного забезпечення. Третій розділ присвячений реалізації проєкту, розробці робочої документації та інструкції. У четвертому розділі розглянуте питання охорони праці.

Основна частина включає 64 сторінок друкованого тексту, 17 рисунків, 2 таблиці, 3 додатки, список використаних джерел містить 17 найменувань.

Робота виконана українською мовою.

Ключові слова: АРМ інвестора, інвестиційний портфель, ІТ- проєкт, управління проектами, автоматизоване робоче місце, Java, SQLite, WebSocket, HTTP сервер, двофакторна автентифікація, вебінтерфейс, графічний інтерфейс, візуалізація даних.

ABSTRACT

The qualification work is devoted to the development of a software system for project management, which will be aimed at investors and business managers. In the first chapter, an analysis of the subject area was carried out, based on the results of the analysis, the concept of the software system was developed and the requirements, task statement and technical task were formulated. In the second section, project solutions for information, software and technical support are developed. The third section is dedicated to project implementation, development of working documentation and instructions. The fourth chapter deals with the issue of labor protection.

The main part includes 64 pages of printed text, 17 drawings, 2 tables, 3 applications, the list of used sources contains 17 titles.

The work was done in Ukrainian.

Keywords: Investor's ARM, investment portfolio, IT project, project management, automated workplace, Java, SQLite, WebSocket, HTTP server, two-factor authentication, web interface, graphical interface, data visualization.

Зміст

Вступ	5
Розділ 1. Розробка концепції автоматизованого робочого місця інвестора ІТ-проекту	7
1.1. Опис об'єкта дослідження та аналіз предметної області	7
1.2. Формування вимог до АРМ інвестора ІТ-проекту	11
1.3. Розробка концепції АРМ інвестора	16
Розділ 2. Розробка проекту інформаційної системи	21
2.1. Загальносистемні рішення	21
2.2. Рішення з інформаційного забезпечення	26
2.3. Рішення з математичного забезпечення	35
2.4. Рішення з програмного забезпечення	38
2.5. Рішення з технічного та програмного забезпечення	41
Розділ 3. Реалізація проекту	44
3.1. Розробка робочої документації	44
3.2. Екранні форми інтерфейсу користувача	47
Розділ 4. Охорона праці при роботі за комп'ютером	57
Висновки	61
Список використаних джерел	63
Додатки	65

Вступ

Актуальність теми дипломного проєкту. Розробка автоматизованого робочого місця (АРМ) інвестора ІТ-проєкту є актуальною через зростаючу потребу в ефективному управлінні інвестиційними портфелями. Це вимагає оперативного доступу до даних, можливості спільної роботи та відстеження змін у режимі реального часу. Проблематика недостатньої автоматизації інвестування, особливо в ІТ – сфері, наразі слабо висвітлена в науковій літературі з точки зору створення комплексних рішень, що обґрунтовує доцільність розробки спеціалізованого інструменту для підтримки діяльності інвестора.

Метою проєкту є створення автоматизованого робочого місця інвестора для управління інвестиційними дошками з підтримкою автентифікації, спільного редагування та моніторингу акцій у режимі реального часу.

Завдання включають розробку концепції інформаційної системи, розробку структури бази даних, реалізація програмного забезпечення та документації для забезпечення ефективної роботи системи.

Об'єктом дослідження є процес управління інвестиційними портфелями в ІТ-проєктах. Предметом є автоматизоване робоче місце інвестора, що забезпечує управління інвестиційними дошками, акціями та маркерами.

Засоби проєктування та створення. Для реалізації проєкту використано мову програмування Java, бібліотеки для роботи з базами даних (SQLite), вебсокетами (Java-WebSocket) та HTTP-сервером (com.sun.net.httpserver). Проєктування здійснювалося з використанням об'єктно-орієнтованого підходу та принципів клієнт-серверної архітектури.

Технічні засоби та програмне забезпечення. Система реалізується на базі серверного застосунку (Server.java) з підтримкою SQLite для зберігання даних, клієнтського інтерфейсу (Client.java) на основі Java Swing, а також веб-інтерфейсу через HTTP-сервер та WebSocket для оновлень у реальному часі.

Використовувалися інструменти для генерації TOTP-кодів для двофакторної автентифікації.

Отримані результати. Розроблено клієнт-серверну систему АРМ інвестора, яка підтримує автентифікацію (включаючи 2FA та OAuth), управління дошками, спільне редагування, відстеження цін акцій у реальному часі та сповіщення. Система включає базу даних, HTTP та WebSocket сервери, а також графічний інтерфейс користувача.

Можливі області застосування. Результати можуть застосовуватися в інвестиційних ІТ-компаніях, фінансовими аналітиками та індивідуальними інвесторами для управління портфелями, спільної роботи та моніторингу ринкових даних.

Обґрунтування необхідності проєктування. Недостатня автоматизація управління інвестиційними портфелями в ІТ-сфері, зокрема відсутність інтегрованих рішень для спільної роботи та моніторингу в реальному часі, зумовлює потребу в створенні автоматизованого робочого місця інвестора. У першому розділі проведено аналіз об'єкта дослідження та сформовано вимоги до системи. У другому розділі описано загальносистемні рішення, структуру бази даних, математичне та програмне забезпечення. У третьому розділі представлено реалізацію системи, включаючи екранні форми та документацію, а в четвертому розділі розглянуто аспекти охорони праці. Розроблена система забезпечує комплексне вирішення проблеми управління інвестиційними портфелями, підвищуючи ефективність роботи інвесторів.

Розділ 1. Розробка концепції автоматизованого робочого місця інвестора ІТ-проєкту

1.1. Опис об'єкта дослідження та аналіз предметної області

Об'єктом дослідження є інвестор ІТ-проєктів — фізична особа або організація, яка вкладає кошти в ІТ-стартапи, продукти чи технології з метою отримання прибутку [1]. Діяльність інвестора в ІТ-сфері є складним і багатогранним процесом, що вимагає аналізу великої кількості даних, оцінки ризиків, прогнозування ринкових тенденцій та прийняття стратегічних рішень. Інвестор в ІТ-проєктах відіграє ключову роль у розвитку технологічного сектору, забезпечуючи фінансування інноваційних ідей, які мають потенціал для масштабування та комерційного успіху [2].

Основна діяльність інвестора ІТ-проєктів включає кілька етапів. На початковому етапі інвестор проводить пошук перспективних проєктів, аналізуючи їх бізнес-моделі, ринковий потенціал, команду, технологічну новизну та конкурентне середовище. Для цього можуть використовуватися різноманітні джерела інформації, такі як стартап-акселератори, краудфандингові платформи, галузеві звіти та мережі контактів. Після відбору потенційних об'єктів інвестування інвестор оцінює фінансові показники проєкту, включаючи прогнозовану дохідність, витрати на розробку, маркетинг та операційну діяльність. Важливим показником є оцінка ризиків, таких як технологічні, ринкові чи регуляторні.

На етапі інвестування інвестор приймає рішення про форму вкладення капіталу: це може бути придбання акцій, пайова участь, надання конвертованих позик або венчурне фінансування [3]. Наприклад, інвестор може придбати частку в ІТ-стартапі через прямий продаж акцій або через спеціалізовані біржі, якщо проєкт уже вийшов на публічний ринок. Після інвестування діяльність інвестора зосереджується на моніторингу проєкту: відстеженні фінансових показників, оцінці прогресу в розробці продукту, аналізі ринкової динаміки та комунікації з

командою стартапу [4]. У деяких випадках інвестор бере активну участь у стратегічному управлінні, надаючи консультації або сприяючи залученню додаткових ресурсів.

Для ефективного управління інвестиційним портфелем інвестору необхідно моделювати різні сценарії розвитку проєктів. Наприклад, моделювання може включати прогнозування зростання вартості акцій на основі ринкових трендів, оцінку впливу зовнішніх факторів (таких як зміни в законодавстві чи економічна нестабільність) або аналіз потенційних сценаріїв виходу з інвестицій (IPO, продаж частки іншому інвестору чи поглинання компанією). Такі моделі часто базуються на статистичних методах, алгоритмах машинного навчання або експертних оцінках.

Автоматизоване робоче місце інвестора IT-проєктів є інформаційною системою, яка підтримує інвестора в управлінні інвестиційним портфелем шляхом інтеграції інструментів для аналізу, прогнозування та комунікації. АРМ забезпечує обробку великих обсягів даних, моделювання сценаріїв розвитку проєктів, відстеження ринкових тенденцій і спільну роботу з іншими учасниками ринку. Наприклад, система дозволяє прогнозувати зростання вартості акцій на основі ринкових трендів, оцінювати вплив зовнішніх факторів (зміни в законодавстві чи економічна нестабільність) або аналізувати потенційні сценарії виходу з інвестицій (IPO, продаж частки іншому інвестору чи поглинання компанією). Такі функції базуються на статистичних методах, алгоритмах машинного навчання або експертних оцінках.

Для виконання своїх завдань інвестор використовує спеціалізовані інструменти, такі як платформи для управління портфелем, аналітичні сервіси для оцінки стартапів, CRM-системи для комунікації та інструменти для моніторингу фінансових ринків. У сучасних умовах швидка реакція на зміни є критично важливою, тому автоматизація процесів управління інвестиціями через АРМ стає необхідною. АРМ інвестора IT-проєктів має забезпечити інтеграцію всіх цих функцій в єдиному середовищі, надаючи зручний інтерфейс для аналізу даних, прийняття рішень і взаємодії з проєктами. Система повинна підтримувати спільну

роботу кількох користувачів, надавати доступ до актуальних ринкових даних, автоматизувати рутинні операції (наприклад, відстеження цін акцій) і забезпечувати безпеку даних, включаючи автентифікацію користувачів і захист інформації.

Існують типові проектні рішення, які можуть використовуватися інвесторами для управління інвестиціями в IT-проектах. Розглянемо їх функціонал і принципи роботи. Для аналізу предметної області розглянемо два популярних рішення: AngelList та Carta.

AngelList — це платформа, яка з'єднує інвесторів зі стартапами, надаючи інструменти для пошуку проєктів, оцінки їх потенціалу та управління інвестиціями [5]. Позитивні аспекти платформи включають широкий доступ до бази стартапів, інтеграцію з венчурними фондами та можливість створення синдикатів для спільного інвестування. AngelList також пропонує аналітичні інструменти для оцінки ринкових трендів і фінансових показників проєктів.

Проте система має низку недоліків. По-перше, AngelList орієнтована переважно на англomовний ринок, що обмежує її використання для інвесторів, які працюють із локальними IT-проектами в інших регіонах. По-друге, платформа стягує високі комісії за транзакції та управління фондами, що може бути фінансово обтяжливим для індивідуальних інвесторів. Крім того, AngelList не надає достатньої гнучкості для налаштування аналітичних інструментів під специфічні потреби користувача, а також має обмежені можливості для прогнозування на основі складних моделей. Відсутність локалізації та підтримки специфічних вимог (наприклад, інтеграції з локальними біржами чи регуляторними органами) робить її менш придатною для сучасних вимог інвесторів в Україні чи інших країнах із унікальними ринковими умовами.

Carta — це платформа для управління активами, яка використовується венчурними фондами та індивідуальними інвесторами для відстеження акцій, управління капіталом і автоматизації звітності [6]. Позитивні сторони Carta включають зручний інтерфейс, інтеграцію з фінансовими системами, підтримку електронного підпису для угод і автоматизацію підготовки звітів для

регуляторних органів. Система також забезпечує безпечне зберігання даних і підтримує управління портфелем у реальному часі.

Недоліки Carta полягають у її високій вартості, що робить її недоступною для індивідуальних інвесторів або малих фондів. Платформа зосереджена на управлінні вже укладеними угодами, але має обмежені можливості для пошуку нових проєктів чи прогнозування їх розвитку. Крім того, Carta не підтримує функції спільного редагування портфелів у реальному часі, що є важливим для командної роботи. Відсутність інструментів для моделювання складних інвестиційних сценаріїв і недостатня адаптація до локальних ринків (зокрема, українських) обмежують її застосування.

Розглянувши ці рішення, можна дійти висновку, що вони не повною мірою вирішують усі завдання, які стоять перед інвестором ІТ-проєктів. Висока вартість, обмежена гнучкість, недостатня локалізація та відсутність інструментів для прогнозування й спільної роботи не дозволяють цим платформам ефективно підтримувати інвесторів, особливо в умовах українського ІТ-ринку. Наприклад, дослідження показують зростання попиту в Україні на доступні платформи для аналізу ринку та групової взаємодії з 2021 року [7]. Обмежена сумісність із API бірж ускладнює імпорт актуальних даних, а затримки в оновленні інформації через нестабільність ринків знижують точність прогнозів [8]. Використання застарілих інструментів, таких як Excel, ускладнює обробку великих обсягів даних і не підтримує реальний час [9]. Крім того, більшість платформ не пропонують локалізованих інтерфейсів для українського ринку, що створює бар'єри для користувачів.

На основі проведеного аналізу можна зробити висновок, що сучасні системи не повною мірою відповідають потребам інвесторів ІТ-проєктів через високу вартість, обмежений функціонал, недостатню інтеграцію з локальними сервісами або слабку підтримку командної роботи. Це створює потребу в розробці нової інформаційної системи у вигляді АРМ, яке буде дешевшим, простішим у використанні, локалізованим для українського ринку та орієнтованим на підвищення ефективності інвестора. АРМ має забезпечувати доступність, швидку

обробку даних, інтеграцію з фінансовими сервісами, підтримку реального часу та функції прогнозування [2, 7].

Обмеження аналізу включають залежність від стабільного інтернет-з'єднання, складність інтеграції з різними фінансовими API, необхідність відповідності стандартам кібербезпеки (GDPR, українське законодавство) та потребу в локалізації для українського ринку [10, 11]. Ці виклики можна подолати завдяки модульному підходу до розробки, використанню WebSocket для синхронізації даних у реальному часі та реалізації сучасних стандартів безпеки [12]. Таким чином, розробка АРМ із урахуванням локальних потреб і сучасних технологій дозволить значно підвищити ефективність інвестора в управлінні ІТ-проєктами.

1.2. Формування вимог до АРМ інвестора ІТ-проєкту

Вебсайт, що є основою автоматизованого робочого місця інвестора ІТ-проєкту, призначений для забезпечення ефективного управління інвестиційними портфелями та аналізу фінансових даних. АРМ створюється на основі вебсайту, який виступає центральною платформою для обробки, зберігання та відображення інформації, пов'язаної з інвестиційною діяльністю. Сайт дозволяє інвесторам обмінюватися інформацією віддалено, підключати інших інвесторів до співпраці та забезпечує зручний інструмент для відстеження ринкових даних, ведення обліку інвестицій, аналізу портфелів і командної роботи.

Сайт має вирішувати наступні ключові завдання:

- Збір і обробка даних передбачає отримання актуальної фінансової інформації, включаючи котирування акцій, ринкові індекси та інші показники, із зовнішніх джерел. Це необхідно для формування аналітичних звітів і прогнозів, що допомагають оцінювати ринкові тенденції.
- Візуалізація інформації реалізується через таблиці, графіки та діаграми, що дозволяють користувачам аналізувати стан портфеля та зміни на ринку. Такий підхід забезпечує наочне представлення фінансових даних.

- Інтерактивність і співпраця дають змогу спільно редагувати портфелі, надавати доступ іншим користувачам із правами редагування або перегляду, а також створювати тимчасові посилання для доступу до даних. Це розширює можливості взаємодії та управління фінансовою інформацією.

- Безпека та автентифікація гарантують захист даних через двофакторну автентифікацію, OAuth і механізми шифрування. Такі заходи забезпечують конфіденційність інформації та захищають її від несанкціонованого доступу.

- Реальний час оновлює ціни акцій та надсилає сповіщення про досягнення цінових порогів за допомогою WebSocket-з'єднання, що забезпечує оперативне реагування на зміни ринку.

Сайт має бути інтуїтивно зрозумілим, з адаптивним дизайном для використання на ПК, планшетах і смартфонах, та підтримувати локалізацію, зокрема українську мову, для забезпечення доступності.

Функціональні вимоги до АРМ інвестора:

1) Формування та ведення інвестиційного портфеля:

- Створення дощок передбачає можливість створення віртуальних дощок, що є аналогами інвестиційних портфелів, із зазначенням активів, їхньої кількості, ціни покупки та ринкової вартості.

- Додавання активів здійснюється через внесення акцій на дошку з вказівкою тікера, кількості, ціни покупки та приміток для обліку активів, що допомагає структуровано вести портфель.

- Оновлення та видалення передбачають можливість редагування інформації про активи або їхнього видалення з портфеля для актуалізації даних відповідно до змін ринку.

- Сповіщення дозволяють налаштовувати повідомлення про досягнення акцією заданої ціни, що забезпечує своєчасне реагування на ринкові коливання.

2) Аналіз портфеля:

- Фінансові розрахунки включають аналіз прибутковості портфеля через порівняння ціни покупки з ринковою вартістю, що дозволяє оцінити ефективність інвестицій.

- Історичні дані забезпечують доступ до хронології змін портфеля, відображаючи, хто і коли вносив корективи, що дозволяє відстежувати історію операцій.

- Візуалізація використовує графічні позначки, такі як стрілки, кола та текстові мітки, що допомагають маркувати інвестиційні стратегії та ключові аналітичні точки.

3) Співпраця та обмін даними:

- Запрошення користувачів передбачає надання доступу до дошки іншим учасникам із можливістю визначення прав редагування або перегляду, що сприяє ефективній командній роботі.

- Створення посилань дозволяє генерувати унікальні посилання з обмеженим терміном дії, надаючи тимчасовий доступ до портфеля без необхідності реєстрації.

- Реальний час забезпечує миттєве відображення змін на дошці, включаючи додавання акцій та позначок, що гарантує актуальність даних для всіх користувачів із доступом.

4) Вирішення інвестиційних задач:

- Моніторинг ринку включає відстеження актуальних цін акцій та ринкових тенденцій, що дає змогу приймати обґрунтовані рішення щодо інвестиційних можливостей.

- Планування стратегій реалізується через створення схем інвестиційних стратегій із використанням графічних позначок і приміток, що дозволяє краще структурувати та аналізувати потенційні варіанти розвитку портфеля.

- Управління ризиками передбачає налаштування сповіщень про цінові пороги, що сприяє мінімізації ризиків завдяки своєчасному реагуванню на ринкові зміни.

Функціональні вимоги до вебсайту:

- Збір і обробка даних передбачає отримання фінансової інформації з API, що використовується для аналізу та прогнозування ринкових змін.
- Візуалізація реалізується через таблиці, графіки та діаграми, що забезпечують зручний спосіб аналізу отриманих даних.
- Інтерактивність дозволяє користувачам спільно редагувати портфелі та надавати доступ кільком користувачам для ефективної взаємодії.
- Безпека гарантується застосуванням двофакторної автентифікації та механізму OAuth, що забезпечує надійний захист даних від несанкціонованого доступу.
- Оновлення в реальному часі здійснюється через WebSocket, що забезпечує миттєву синхронізацію даних та сповіщення про важливі зміни.

Вимоги до програмного забезпечення для АРМ:

- Мова програмування: високорівнева мова для реалізації логіки клієнт-серверної взаємодії.
- База даних: система управління базами даних для зберігання інформації про користувачів, портфелі та історію змін.
- Вебфреймворк: інструмент для створення вебінтерфейсу та обробки HTTP-запитів.
- Бібліотеки для роботи з API: інструменти для інтеграції з фінансовими API.
- Бібліотеки безпеки: модулі для реалізації шифрування та автентифікації.

Вимоги до програмного забезпечення для вебсайту:

- Мова програмування: високорівнева мова для створення серверної та клієнтської частин.
- Система управління базами даних: для зберігання даних користувачів, портфелів і транзакцій.
- Вебфреймворк: для обробки запитів і відображення інтерфейсу.

- WebSocket: для реалізації оновлень у реальному часі.
- Бібліотеки безпеки: для захисту даних і автентифікації користувачів.

Вимоги до апаратного забезпечення для АРМ:

- Персональний комп'ютер: сучасна операційна система, процесор рівня Intel Core i5, 8 ГБ оперативної пам'яті, монітор із роздільною здатністю 1920×1080.

- Інтернет-з'єднання: стабільне з'єднання зі швидкістю не менше 10 Мбіт/с.

- Мобільні пристрої: смартфон Android для віддаленого доступу.
- Додаткове обладнання: миша, клавіатура, принтер для друку звітів.

Вимоги до апаратного забезпечення для вебсайту:

- Сервер: сучасна операційна система, процесор рівня Intel Xeon, 16 ГБ оперативної пам'яті.

- Інтернет-з'єднання: високошвидкісне з'єднання для обробки запитів користувачів.

- Хмарне сховище: для резервного копіювання даних і масштабування.

Сервісні засоби для АРМ:

- Електронна пошта: для надсилання сповіщень і підтверджень.
- Захист даних: шифрування та двофакторна автентифікація для безпеки.

- Локалізація: підтримка української мови в інтерфейсі.

- Технічна підтримка: система для обробки запитів користувачів.

Сервісні засоби для вебсайту:

- Електронна пошта: для сповіщень і підтвердження доступу.
- Захист: шифрування, двофакторна автентифікація та OAuth.
- Локалізація: інтерфейс українською мовою.
- Моніторинг: інструменти для відстеження працездатності сервера.

На основі сформульованих вимог буде розроблена концепція АРМ інвестора ІТ-проєкту, яка поєднує функціональність вебсайту з можливостями

управління портфелем, аналізу даних і співпраці. Функціональні вимоги забезпечують ефективне вирішення інвестиційних задач, а технічні та сервісні вимоги гарантують стабільну роботу системи на сучасному обладнанні з мінімальними ресурсами, роблячи АРМ універсальним і зручним інструментом для інвесторів.

1.3. Розробка концепції АРМ інвестора

На основі сформованих вимог до автоматизованого робочого місця інвестора ІТ-проєкту розроблено детальну концепцію системи, яка забезпечує часткову автоматизацію процесів управління інвестиціями, аналізу їхньої ефективності та взаємодії між інвесторами. Ця концепція ґрунтується на моделі клієнт-серверної архітектури, яка включає інтеграцію з фінансовими системами, обробку даних у реальному часі та зручний графічний інтерфейс для користувачів. Модель концепції представлена на рисунку 1.1.

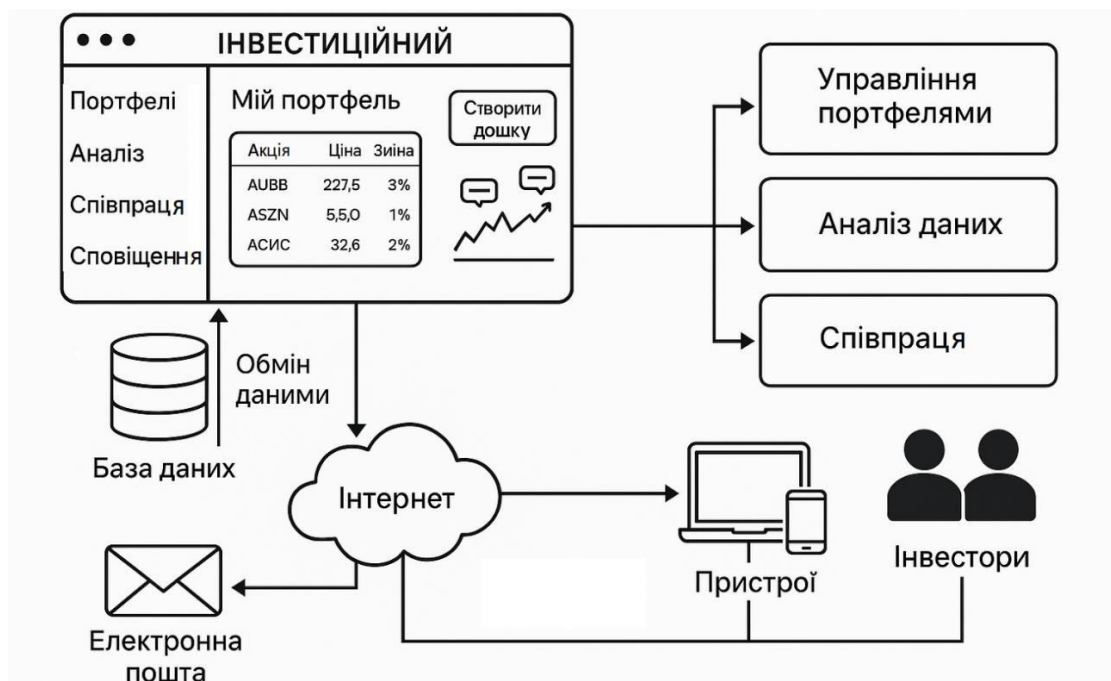


Рисунок 1.1 – Модель концепції автоматизованого робочого місця інвестора ІТ-проєкту

Концепція вебсайту, що є основою АРМ, передбачає створення платформи, доступної через інтернет, для інвесторів, які мають стабільне з'єднання. Сайт

дозволяє інвесторам обмінюватися інформацією, пересилати дані про портфелі через посилання з обмеженим терміном дії, запрошувати інших користувачів до співпраці та відстежувати ринкові дані в реальному часі. Основні задачі сайту включають:

- Управління портфелями, яке включає створення та редагування інвестиційних дощок із можливістю додавання акцій, позначок та приміток, що забезпечує структуроване ведення фінансових активів.

- Аналіз даних, що здійснюється через візуалізацію ринкових показників у таблицях та графічних позначках, що дозволяє оцінювати ефективність портфеля.

- Співпраця, що передбачає надання доступу до портфеля іншим інвесторам із можливістю редагування або перегляду, що сприяє командному управлінню активами.

- Сповіщення, які забезпечують оновлення цін акцій і надсилання повідомлень про досягнення цінових порогів у реальному часі, що дозволяє оперативно реагувати на зміни ринку.

- Безпека, що гарантується через використання двофакторної автентифікації, OAuth і шифрування, що захищає дані від несанкціонованого доступу.

Інтерфейс сайту локалізований українською мовою, адаптивний для різних пристроїв (ПК, планшети, смартфони) і забезпечує інтуїтивну навігацію. Інвестори можуть входити на сайт через автентифікацію, створювати дошки, ділитися ними через посилання та отримувати сповіщення про ринкові зміни.

Концепція АРМ інвестора базується на вебсайті та включає клієнтську програму з графічним інтерфейсом, реалізовану в Client.java, і серверну частину, реалізовану в Server.java. АРМ забезпечує інвесторам зручний доступ до функцій сайту через настільний додаток із підтримкою локалізації українською мовою. Клієнтська частина дозволяє:

- виконувати автентифікацію (включаючи 2FA та OAuth);
- створювати та управляти інвестиційними дошками;

- додавати акції з інформацією про тікер, кількість, ціну покупки та примітки;
- розміщувати графічні позначки (стрілки, кола, текст) для аналізу стратегій;
- переглядати історію змін дошки.

Серверна частина обробляє запити клієнтів, забезпечує безпеку (хешування паролів, управління сесіями, CORS-заголовки), оновлює ціни акцій кожні 15 хвилин, підтримує WebSocket для сповіщень у реальному часі та керує доступом через посилання. База даних, реалізована на SQLite, зберігає дані про користувачів, дошки, акції, позначки, історію змін і посилання для спільного доступу, з можливістю міграції на PostgreSQL для масштабування.

Модель концепції передбачає модульну структуру, що дозволяє розширювати функціонал, наприклад, додаванням підтримки криптовалют, аналітичних графіків або інтеграції з API, такими як Alpha Vantage. Код Server.java демонструє гнучкість системи через методи для оновлення цін акцій, перевірки сповіщень і очищення прострочених сесій. Client.java забезпечує зручний інтерфейс із таблицями для акцій і панеллю для позначок, що відповідає вимогам до візуалізації та інтерактивності.

Інтеграція з фінансовими системами передбачає використання сервісу Stripe для обробки платежів, що дозволяє інвесторам безпечно вносити кошти. Дані транзакцій зберігаються в базі даних, а підтвердження надсилаються через сервіс Mailgun у форматі PDF. Server.java підтримує обробку HTTP-запитів для таких операцій і створення посилань для залучення нових інвесторів. Для ринкових даних у майбутньому планується інтеграція з API Yahoo Finance або Alpha Vantage, що замінить симуляцію в методі updateStockPrices() на реальні запити.

Безпека залишається пріоритетом. Server.java реалізує:

- хешування паролів за допомогою SHA-256;
- двофакторну автентифікацію через TOTP;
- OAuth-автентифікацію через Google;
- управління сесіями з очищенням кожні 30 хвилин;

- захист від міжсайтових атак через CORS-заголовки.

Ці механізми можуть бути доповнені шифруванням даних у базі даних і використанням HTTPS.

Масштабованість забезпечується модульною структурою `Server.java`, яка дозволяє додавати нові команди, наприклад, для аналітичних звітів у PDF через бібліотеку `iText`. `Client.java` може бути розширений для відображення звітів або чату між інвесторами через `WebSocket`, що підвищить інтерактивність.

Програмні ресурси для технічної реалізації:

- Мова програмування: Java для серверної та клієнтської частин.
- База даних: `SQLite` із можливістю переходу на `PostgreSQL`.
- Вебінтерфейс: `HTML`, `CSS`, `JavaScript` для обробки `HTTP`-запитів.
- Графічний інтерфейс: `Swing` для кросплатформного клієнтського додатка.
- Інструменти документації: `PlantText`, `Microsoft Word`, `Paint` для діаграм і матеріалів.

Ці ресурси забезпечують створення гнучкої та масштабованої системи. У майбутньому можливий перехід на `Spring` для серверної частини або `React` для вебінтерфейсу для спрощення розробки та підвищення продуктивності.

Розроблена концепція АРМ інвестора забезпечує комплексне рішення для автоматизації інвестиційних процесів. Код `Server.java` та `Client.java` реалізує ключові функції: автентифікацію, управління дошками, обробку акцій і позначок, сповіщення в реальному часі. Модульна архітектура дозволяє додавати нові функції, такі як аналітичні звіти, чат або інтеграцію з API. Концепція відповідає вимогам до зручності, безпеки, масштабованості та інтеграції з фінансовими системами, створюючи сучасний інструмент для інвесторів, готовий до подальшого розвитку.

Отже, автоматизоване робоче місце інвестора IT-проектів є необхідним інструментом для ефективного прийняття інвестиційних рішень в умовах динамічного ринку. Його впровадження дозволить оптимізувати процеси аналізу,

моніторингу та управління інвестиціями, підвищуючи точність і оперативність дій інвестора.

У цьому розділі було сформовано концепцію АРМ інвестора, обґрунтовано необхідність його створення та визначено функціональні вимоги до системи, що забезпечують ефективне управління інвестиціями в ІТ-сфері.

Розділ 2. Розробка проєкта інформаційної системи

2.1. Загальносистемні рішення

Розробка автоматизованого робочого місця інвестора ІТ-проєкту, представлена у наданому коді, є прикладом створення інформаційної системи, що забезпечує ефективне управління інвестиційними дошками з підтримкою автентифікації, спільного редагування та відстеження змін у реальному часі. У цьому розділі розглядаються загальносистемні рішення, які охоплюють схему організаційної структури, функціональну структуру, опис функцій, моделі предметної області та загальний опис системи. Ці аспекти відповідають вимогам до проєктування інформаційних систем, забезпечуючи комплексний підхід до автоматизації діяльності інвесторів.

Організаційна структура системи автоматизованого робочого місця інвестора передбачає взаємодію між різними учасниками, які забезпечують функціонування системи або використовують отриману інформацію для прийняття рішень. Основними учасниками є:

- Користувачі (інвестори): фізичні особи або організації, які створюють, редагують і переглядають інвестиційні дошки, додають акції, встановлюють сповіщення про ціни та додають графічні позначки для аналізу.
- Адміністратори системи: відповідають за підтримку серверної інфраструктури, бази даних та забезпечення безпеки, включаючи автентифікацію та захист даних.
- Зовнішні сервіси: системи, що надають дані про ціни акцій (наприклад, Yahoo Finance), а також сервіси автентифікації (Google OAuth для входу через Google).

Основні функції та зв'язки між цими учасниками полягають у створенні та управлінні дошками, обміні інформацією через клієнт-серверну архітектуру та забезпеченні безпеки доступу. Користувачі взаємодіють із системою через клієнтський додаток, який передає запити до сервера. Адміністратори

забезпечують безперебійну роботу сервера та бази даних, а зовнішні сервіси надають додаткові дані, необхідні для функціонування системи. Підпорядкованість у системі відсутня, оскільки користувачі мають рівні права в межах своїх доступів до дошок, а адміністратори виконують технічну підтримку.

Функціональна структура системи складається з кількох ключових компонентів, які забезпечують виконання автоматизованих функцій. Основними елементами є:

1) Підсистеми:

- Серверна підсистема: обробляє запити клієнтів, управляє базою даних, забезпечує автентифікацію (включаючи двофакторну автентифікацію та OAuth) і реалізує WebSocket для оновлень у реальному часі.

- Клієнтська підсистема: графічний інтерфейс користувача (GUI), який дозволяє взаємодіяти з дошками, акціями та позначками.

- База даних: SQLite база для зберігання інформації про користувачів, дошки, акції, позначки, сесії, історію змін та посилання для спільного доступу.

- WebSocket сервер: забезпечує реальнечасові оновлення та сповіщення.

2) Комплекси задач:

- Управління користувачами: реєстрація, вхід, вихід, налаштування 2FA, вхід через Google.

- Управління дошками: створення, редагування, запрошення користувачів, створення посилань для спільного доступу.

- Управління акціями: додавання, оновлення, видалення акцій, встановлення сповіщень про ціни.

- Управління позначками: додавання, оновлення, видалення графічних позначок на дошках.

- Оновлення цін акцій: періодичне оновлення цін через зовнішні API та перевірка умов сповіщень.

- Історія змін: збереження та перегляд історії дій на дошках.

3) Інформаційні зв'язки:

- Клієнт-сервер: клієнт надсилає запити (наприклад, “LOGIN”, “CREATE_BOARD”) у текстовому форматі через сокети, а сервер повертає відповіді (наприклад, “SUCCESS|sessionId”).
- Сервер-база даних: сервер виконує SQL-запити для збереження та отримання даних.
- WebSocket зв’язки: сервер надсилає повідомлення про оновлення дощок або сповіщення клієнтам, підключеним до відповідних дощок.
- Зовнішні зв’язки: сервер періодично звертається до API цін акцій для оновлення даних.

Розробка системи базується на таких вихідних матеріалах:

- Специфікація вимог до автоматизованого робочого місця інвестора, включаючи підтримку автентифікації, спільного редагування та реальногочасових оновлень.
- Документація Java SE, SQLite, Java-WebSocket та бібліотек для роботи з HTTP (com.sun.net.httpserver).
- Стандарти безпеки для автентифікації (SHA-256 для хешування паролів, TOTP для 2FA, Google OAuth).

Особливості об’єкта управління, які вплинули на проєктні рішення:

- Необхідність підтримки спільного редагування кількома користувачами.
- Вимога до безпеки доступу через двофакторну автентифікацію та OAuth.
- Потреба в реальномучасових оновленнях для відображення змін на дошках.

Інтерфейси системи включають:

- Клієнтський GUI: створений за допомогою Java Swing, забезпечує інтерактивний доступ до функцій.
- Сокетний інтерфейс: текстові запити через TCP-сокети.
- HTTP API: обробка запитів для веб-інтерфейсу.

- WebSocket: для передачі реальногочасових оновлень.

Інтерфейсна інформація

Система імпортує:

- дані про ціни акцій із зовнішніх API (наприклад, Yahoo Finance);
- дані користувачів через Google OAuth для автентифікації.

Експортує:

- повідомлення та сповіщення через WebSocket;
- дані дощок, акцій і позначок для відображення клієнтам;
- історію змін для аналізу дій користувачів.

Характеристика функціональної структури

Інформаційна модель системи включає сервер, який обробляє запити, базу даних для зберігання інформації та клієнтський додаток для взаємодії користувачів. Основні задачі:

- Автентифікація: обробка входу, реєстрації, 2FA та OAuth.
- Управління дошками: створення, запрошення, спільний доступ.
- Обробка акцій: додавання, оновлення, видалення, сповіщення.
- Обробка позначок: додавання, редагування, видалення графічних елементів.

- Реальногочасові оновлення: через WebSocket.

Процес виконання функцій автоматизований (обробка запитів, оновлення цін) або частково автоматизований (введення даних користувачем через GUI).

Вимоги до реалізації:

- Точність: коректне відображення цін акцій і змін дощок.
- Надійність: стійкість до збоїв мережі та бази даних.
- Швидкість: оновлення цін кожні 15 хвилин, реальногочасові оновлення через WebSocket.

Використані типові програмні засоби:

- Java SE: для серверної та клієнтської логіки.
- SQLite: для зберігання даних.

- Java-WebSocket: для реальногочасових оновлень.
- Swing: для створення GUI.
- com.sun.net.httpserver: для HTTP API.

Ці засоби застосовані для всіх модулів системи, включаючи автентифікацію, управління дошками та обробку акцій.

Загальне опис системи

Система призначена для автоматизації діяльності інвесторів, зокрема управління інвестиційними портфелями через створення та редагування дощок. Вона використовується на об'єктах автоматизації, таких як персональні комп'ютери або сервери, де інвестори аналізують ринкові дані. Основні функції:

- реєстрація та автентифікація користувачів;
- створення та управління дошками;
- додавання, оновлення та видалення акцій;
- встановлення сповіщень про ціни акцій;
- додавання графічних позначок;
- спільне редагування та реальногочасові оновлення.

Система має клієнт-серверну архітектуру:

- Сервер: обробляє запити, управляє базою даних, забезпечує WebSocket та HTTP API.
- Клієнт: Java Swing додаток із GUI для взаємодії з дошками.
- База даних: SQLite для зберігання даних користувачів, дощок, акцій, позначок та історії.

Функціонування системи включає ініціалізацію сервера, обробку клієнтських запитів, періодичне оновлення цін акцій і передачу реальногочасових оновлень через WebSocket.

Система пов'язана з:

- Google OAuth: для автентифікації користувачів.
- API цін акцій: для отримання ринкових даних.

Зв'язки реалізуються через HTTP-запити до Google API для валідації токенів та запити до зовнішніх API для оновлення цін акцій.

Регламент зв'язків:

- Періодичне оновлення цін кожні 15 хвилин.
- Реальногочасові WebSocket повідомлення при змінах на дошках.

Розроблена система забезпечує комплексне рішення для автоматизації робочого місця інвестора, поєднуючи функціональність управління портфелями, безпечну автентифікацію та реальногочасові оновлення. Використання стандартних технологій Java, SQLite та WebSocket забезпечує надійність і масштабованість, а підтримка спільного редагування та сповіщень відповідає сучасним вимогам до інвестиційних платформ.

Створена концептуальна модель представлена на рисунку 2.1.

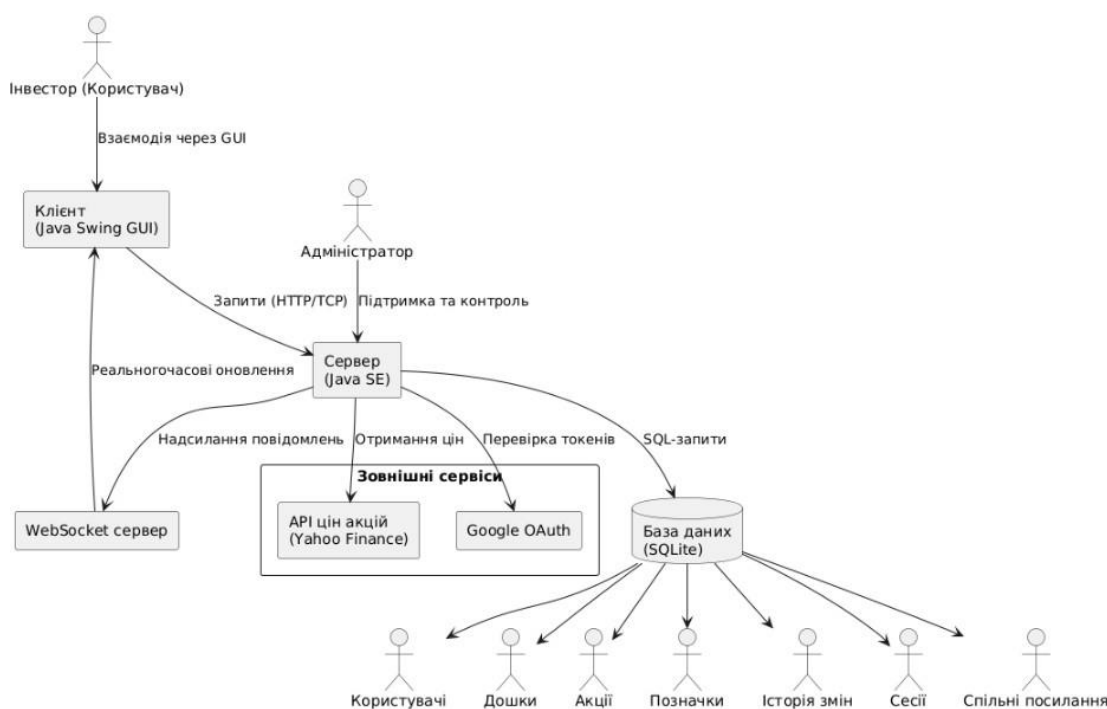


Рисунок 2.1 – Концептуальна модель

2.2. Рішення з інформаційного забезпечення

Інформаційне забезпечення автоматизованого робочого місця інвестора IT-проекту є ключовим елементом, що забезпечує ефективне функціонування

системи, її взаємодію з користувачами та обробку даних. У даному підрозділі розглядаються складові інформаційного забезпечення, принципи його організації, системи класифікації та кодування інформації, структура інформаційної бази, а також інструкція щодо її формування та ведення. Нижче детально описано кожен з аспектів відповідно до вимог.

Склад інформаційного забезпечення

Інформаційна система автоматизованого робочого місця інвестора побудована на основі реляційної бази даних SQLite [13], що міститься у файлі investor.db. Основна база даних включає кілька ключових таблиць, кожна з яких виконує специфічні функції (рисунок 2.2):

- Таблиця users: зберігає інформацію про користувачів системи, включаючи їх email, пароль, налаштування двофакторної автентифікації (2FA), дані OAuth та час останнього входу.
- Таблиця boards: містить дані про інвестиційні дошки, включаючи їх назву, власника, дату створення та останньої модифікації, а також колір фону.
- Таблиця user_board: визначає права доступу користувачів до дощок (перегляд або редагування).
- Таблиця stocks: зберігає інформацію про акції, додані до дощок, включаючи символ акції, кількість, ціну купівлі, поточну ціну, примітки та налаштування сповіщень.
- Таблиця markers: містить дані про графічні позначки на дошках, такі як тип, координати, опис, колір та розмір.
- Таблиця share_links: призначена для зберігання посилань для спільного доступу до дощок, включаючи токен доступу, термін дії та права редагування.
- Таблиця board_history: фіксує історію змін дощок для відстеження дій користувачів.
- Таблиця sessions: зберігає інформацію про активні сесії користувачів, включаючи ідентифікатор сесії, час створення, термін дії та IP-адресу клієнта.

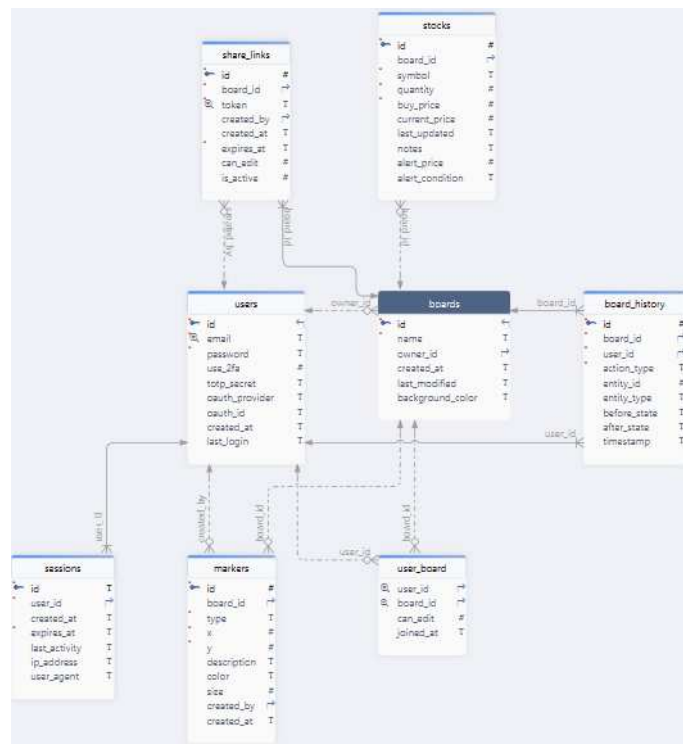


Рисунок 2.2 – Структура бази даних

Кожна таблиця має чітке призначення, що дозволяє системі ефективно керувати користувачами, дошками, акціями та їх взаємозв'язками.

Інформаційне забезпечення системи базується на реляційній моделі даних, що забезпечує структуроване зберігання інформації та швидкий доступ до неї. Основними принципами є:

- **Модульність**: дані розподілені між таблицями за функціональними ознаками, що спрощує їх обробку та модифікацію.
- **Цілісність**: використання зовнішніх ключів (FOREIGN KEY) забезпечує зв'язність між таблицями, наприклад, між boards і users або stocks і boards.
- **Доступність у реальному часі**: система підтримує оновлення даних у реальному часі через WebSocket-з'єднання, що дозволяє синхронізувати зміни між користувачами.
- **Безпека**: двофакторна автентифікація, хешування паролів та обмеження доступу за сесіями забезпечують захист даних.

Для зберігання даних обрано SQLite як основний носій інформації через його легковагість, простоту інтеграції та відсутність потреби в окремому сервері

бази даних. SQLite підходить для проєктів середнього масштабу, таких як автоматизоване робоче місце інвестора, оскільки забезпечує швидкий доступ до даних і не вимагає складного адміністрування. Дані розподілені між таблицями для забезпечення логічної структуризації: статичні дані (користувачі, дошки) зберігаються окремо від динамічних (акції, позначки), що оптимізує швидкість обробки запитів. Додатково, кешування цін акцій у пам'яті (stockPriceCache) зменшує навантаження на базу даних при частих запитах.

Контроль даних здійснюється на кількох рівнях:

- Цілісність даних: зовнішні ключі та унікальні обмеження (наприклад, унікальність email у таблиці users) запобігають дублюванню та некоректним зв'язкам.
- Перевірка введення: сервер перевіряє коректність числових параметрів (наприклад, кількість акцій чи координати позначок) перед внесенням у базу.
- Логування подій: таблиця board_history фіксує всі зміни (створення, редагування, видалення), що дозволяє відстежувати дії користувачів.
- Автоматичне очищення: заплановані завдання (scheduleMaintenanceTasks) видаляють прострочені сесії та посилання для доступу, підтримуючи актуальність даних.

Застосування системи класифікації та кодування інформації

У системі застосована класифікація об'єктів, що відповідає функціональним потребам інвестиційної платформи:

- Користувачі: класифікуються за email (унікальний ідентифікатор) та типом автентифікації (звичайна, 2FA, OAuth).
- Дошки: класифікуються за власником та назвою, з унікальним числовим ідентифікатором (id).
- Акції: класифікуються за символом акції (наприклад, AAPL) та дошкою, до якої вони належать.
- Позначки: класифікуються за типом (ARROW, CIRCLE, TEXT) та координатами на дошці.

- Посилання для доступу: класифікуються за токеном та терміном дії.

Така класифікація забезпечує чітке розмежування об'єктів і полегшує їх обробку в системі.

Для кодування об'єктів використовуються наступні підходи:

- Числові ідентифікатори: кожна таблиця має унікальний ідентифікатор (id), що генерується автоматично (AUTOINCREMENT).
- Текстові ідентифікатори: символи акцій (наприклад, AAPL, TSLA) використовуються як стандартні біржові коди.
- Унікальні токени: для сесій (sessions) та посилань для доступу (share_links) генеруються UUID, що забезпечує унікальність і безпеку.
- Хешування: паролі користувачів хешуються за допомогою алгоритму SHA-256 для забезпечення безпеки.
- TOTP для 2FA: для двофакторної автентифікації використовується алгоритм TOTP, де секретний ключ кодується у форматі Base64.

Ці засоби кодування забезпечують унікальність, безпеку та зручність обробки даних.

Організація інформаційної бази

Інформаційна база побудована на основі реляційної моделі даних, що забезпечує структуроване зберігання та швидкий доступ до інформації. Основні принципи:

- Нормалізація: дані розподілені між таблицями для уникнення надмірності (наприклад, інформація про користувачів і дошки зберігається окремо, а зв'язок встановлюється через user_board).
- Масштабованість: SQLite дозволяє легко розширювати базу шляхом додавання нових таблиць або полів.
- Доступність у реальному часі: WebSocket-сервер забезпечує миттєве оновлення даних для всіх підключених клієнтів.

Обсяг бази даних залежить від кількості користувачів, дощок та акцій. Наприклад, таблиця stocks може містити тисячі записів для активних

користувачів, тоді як таблиця `users` зазвичай обмежується кількістю зареєстрованих осіб.

Структура бази даних включає вісім основних таблиць, пов'язаних через зовнішні ключі:

- `users`: пов'язана з `boards` (власник дошки), `user_board` (права доступу), `sessions` (сесії користувачів), `share_links` (створення посилань) та `board_history` (дії користувача).
- `boards`: пов'язана з `stocks`, `markers`, `user_board`, `share_links` та `board_history` через поле `board_id`.
- `user_board`: забезпечує зв'язок між `users` і `boards`, визначаючи права доступу.
- `stocks`: пов'язана з `boards` через `board_id`, містить дані про акції.
- `markers`: також пов'язана з `boards` через `board_id`, містить дані про позначки.
- `share_links`: пов'язана з `boards` та `users` (створювач посилання).
- `board_history`: фіксує зміни, пов'язані з `boards` та `users`.
- `sessions`: пов'язана з `users` через `user_id`.

Ці зв'язки забезпечують цілісність даних і дозволяють ефективно виконувати запити.

Нормативно-довідкова інформація включає:

- Символи акцій: зберігаються в таблиці `stocks` і використовуються для отримання цін через зовнішні API (у коді симулюється).
- Типи позначок: фіксований набір типів (`ARROW`, `CIRCLE`, `TEXT`) у таблиці `markers`.
- Права доступу: визначаються в таблиці `user_board` (0 для перегляду, 1 для редагування).

Ця інформація оновлюється автоматично (наприклад, ціни акцій) або вручну користувачами (наприклад, додавання нових позначок).

Опис інформаційної бази

Логічна структура бази даних складається з таблиць, описаних вище, кожна з яких має чітко визначені поля [14]:

- users: id (INTEGER, PK), email (TEXT, унікальний), password (TEXT), use_2fa (INTEGER), totp_secret (TEXT), oauth_provider (TEXT), oauth_id (TEXT), created_at (TEXT), last_login (TEXT).
- boards: id (INTEGER, PK), name (TEXT), owner_id (INTEGER, FK), created_at (TEXT), last_modified (TEXT), background_color (TEXT).
- user_board: user_id (INTEGER, FK), board_id (INTEGER, FK), can_edit (INTEGER), joined_at (TEXT).
- stocks: id (INTEGER, PK), board_id (INTEGER, FK), symbol (TEXT), quantity (INTEGER), buy_price (REAL), current_price (REAL), last_updated (TEXT), notes (TEXT), alert_price (REAL), alert_condition (TEXT).
- markers: id (INTEGER, PK), board_id (INTEGER, FK), type (TEXT), x (INTEGER), y (INTEGER), description (TEXT), color (TEXT), size (INTEGER), created_by (INTEGER, FK), created_at (TEXT).
- share_links: id (INTEGER, PK), board_id (INTEGER, FK), token (TEXT, унікальний), created_by (INTEGER, FK), created_at (TEXT), expires_at (TEXT), can_edit (INTEGER), is_active (INTEGER).
- board_history: id (INTEGER, PK), board_id (INTEGER, FK), user_id (INTEGER, FK), action_type (TEXT), entity_id (INTEGER), entity_type (TEXT), before_state (TEXT), after_state (TEXT), timestamp (TEXT).
- sessions: id (TEXT, PK), user_id (INTEGER, FK), created_at (TEXT), expires_at (TEXT), last_activity (TEXT), ip_address (TEXT), user_agent (TEXT).

Формати даних включають числові (INTEGER, REAL), текстові (TEXT) та булеві (INTEGER для 0/1). Взаємозв'язки між таблицями забезпечуються через зовнішні ключі, що гарантують цілісність даних.

Фізично база даних зберігається у файлі investor.db на диску серверної машини. SQLite використовує єдиний файл для всіх таблиць, що спрощує резервне копіювання та перенесення даних. Доступ до даних здійснюється через

JDBC-драйвер, а кешування цін акцій (stockPriceCache) реалізовано в оперативній пам'яті для зменшення кількості запитів до диску.

Ведення бази даних включає:

- Автоматичне створення: при запуску сервера виконується ініціалізація бази даних (initDatabase), яка створює всі необхідні таблиці.
- Оновлення даних: ціни акцій оновлюються кожні 15 хвилин (updateStockPrices), а прострочені сесії та посилання видаляються за розкладом.
- Логування змін: кожна дія (створення дошки, додавання акції, запрошення користувача) фіксується в таблиці board_history.
- Захист даних: паролі хешуються, доступ до даних обмежується через перевірку сесій, а TOTP забезпечує додатковий рівень безпеки для 2FA.

Зовнішня база включає документи, такі як запрошення користувачів (електронні листи, згенеровані системою) та посилання для спільного доступу, які передаються користувачам через WebSocket або HTTP. Маршрути руху документів включають:

- Відправлення запрошень користувачам через email (симулюється в коді).
- Передача посилань для доступу через зовнішні канали (наприклад, електронна пошта або месенджери).

Інструкція з формування та ведення бази даних

Правила підготовки даних.

1) Відбір інформації: дані для внесення до бази (наприклад, символи акцій, координати позначок) вводяться користувачами через графічний інтерфейс клієнта.

2) Кодування інформації: символи акцій відповідають біржовим стандартам (наприклад, AAPL), паролі хешуються за допомогою SHA-256, а сесії та посилання кодуються через UUID.

3) Форми представлення: дані вводяться через текстові поля та випадаючі списки у клієнтському інтерфейсі, з перевіркою форматів (наприклад, числові значення для кількості акцій).

4) Внесення змін: зміни вносяться через запити до сервера (наприклад, `ADD_STOCK`, `UPDATE_MARKER`), з попередньою валідацією даних.

Порядок і засоби заповнення бази даних.

Технічні засоби: клієнтський додаток (Java Swing) відправляє запити до сервера через сокетне з'єднання або HTTP API. Сервер обробляє запити через JDBC-драйвер для SQLite.

Процедура заповнення:

1) Користувач вводить дані через графічний інтерфейс (наприклад, символ акції, кількість).

2) Клієнт формує запит (наприклад, `ADD_STOCK|sessionId|boardId|symbol|quantity|price|notes`).

3) Сервер перевіряє права доступу та коректність даних, після чого вносить їх до бази через SQL-запит (`INSERT INTO stocks ...`).

4) Підтвердження успішного виконання повертається клієнту.

Процедура зміни та контролю бази даних.

Зміна даних: оновлення даних (наприклад, ціни акцій чи координат позначок) виконується через запити `UPDATE` з перевіркою прав доступу (`can_edit` у таблиці `user_board`).

Контроль даних:

- Перевірка цілісності через зовнішні ключі.
- Логування всіх змін у таблиці `board_history`.
- Автоматичне очищення прострочених даних (`cleanExpiredSessions`, `cleanExpiredShareLinks`).

Порядок і засоби відновлення бази даних.

- Захист від руйнування: SQLite забезпечує атомарність транзакцій, що запобігає пошкодженню даних при збоях.
- Резервне копіювання: періодичне копіювання файлу `investor.db` (реалізується поза кодом, але може бути автоматизовано).
- Захист від несанкціонованого доступу: доступ до бази обмежується перевіркою сесій та 2FA.

– Відновлення: у разі пошкодження бази даних її можна відновити з резервної копії, після чого виконується повторна ініціалізація (initDatabase).

Ця структура інформаційного забезпечення забезпечує надійне зберігання, обробку та захист даних, необхідних для роботи автоматизованого робочого місця інвестора, відповідаючи вимогам ефективності, безпеки та зручності використання.

2.3. Рішення з математичного забезпечення

Розробка автоматизованого робочого місця інвестора IT-проєкту, представлена у наданих лістингах кодів Server.java та Client.java, базується на комплексному математичному забезпеченні, яке включає алгоритми обробки даних, методи моделювання та побудову математичних моделей для управління інвестиційними дошками. Нижче розкрито основні аспекти математичного забезпечення, включаючи алгоритми, методи моделювання та контроль достовірності обчислень.

Основним алгоритмом, реалізованим у системі, є алгоритм обробки запитів клієнта на сервері для виконання операцій з інвестиційними дошками, такими як створення дощок, додавання акцій, встановлення сповіщень та управління маркерами. Логіка алгоритму базується на клієнт-серверній взаємодії через сокети та HTTP-запити. Запити клієнта (наприклад, LOGIN, CREATE_BOARD, ADD_STOCK) обробляються сервером у послідовності:

- Парсинг запиту: запит розбивається на команду та параметри (наприклад, command|param1|param2).
- Перевірка сесії: Для кожної операції перевіряється валідність сесії користувача через sessionId у таблиці sessions.
- Виконання операції: залежно від команди виконується відповідна дія (наприклад, вставка даних у базу SQLite для створення дошки або оновлення цін акцій).

– Формування результату: сервер повертає відповідь у форматі SUCCESS|data або ERROR|message.

Для оновлення цін акцій використовується алгоритм періодичного оновлення (updateStockPrices), який виконується кожні 15 хвилин. Він включає:

- отримання унікальних символів акцій із таблиці stocks.
- моделювання коливань цін у межах $\pm 5\%$ від попередньої ціни з округленням до двох знаків після коми можна розрахувати за допомогою формули 2.3.1.

$$currentPrice = \max(0.01, lastPrice + (lastPrice \cdot (random \cdot 0.10 - 0.05))) \quad (2.3.1)$$

– Оновлення цін у базі даних та перевірка умов сповіщень (checkPriceAlerts).

У системі застосовується метод комп'ютерного моделювання, який базується на дискретно-подійному підході. Кожна дія користувача (наприклад, додавання акції чи маркера) моделюється як подія, що викликає зміну стану бази даних та розсилку оновлень через WebSocket. Модель даних включає таблиці (users, boards, stocks, markers), які відображають структуру інвестиційних дощок. Для реального часу використовується WebSocket-сервер, який забезпечує асинхронне оновлення даних для всіх підключених клієнтів.

Метод моделювання цін акцій спрощує реальні ринкові процеси, використовуючи псевдовипадкові коливання цін, що дозволяє уникнути складних фінансових моделей, таких як Блека-Шоулза, але забезпечує достатню точність для демонстраційних цілей. Критерієм вибору методу є низька обчислювальна складність та можливість інтеграції з зовнішніми API (наприклад, Yahoo Finance) у майбутньому.

Математична модель системи складається з таких компонентів:

1) Модель користувача: Користувач ідентифікується через id, email, password та sessionId. Сесії моделюються з терміном дії (24 години), розрахунок проводиться за допомогою формули 2.3.2.

$$expiresAT = currentTime + 24 \cdot 3600 \text{ (секунд)} \quad (2.3.2)$$

2) Модель дошки: Дошка визначається параметрами `id`, `name`, `owner_id`, `background_color`. Доступ користувачів моделюється через таблицю `user_board` з булевою змінною `can_edit`.

3) Модель акцій: Акція характеризується параметрами `symbol`, `quantity`, `buy_price`, `current_price`. Сповіщення моделюються через умову, за допомогою формули 2.3.3.

$$\begin{aligned} & alertTriggerer \\ &= \{ \begin{array}{l} true, if(condition = "ABOVE" \wedge currentPrice \geq alertPrice) \vee \\ (condition = "BELOW" \wedge currentPrice \leq alertPrice) \\ false, otherwise \end{array} \quad (2.3.3) \end{aligned}$$

4) Модель маркерів: Маркери описуються координатами (`x`, `y`), типом (`ARROW`, `CIRCLE`, `TEXT`) та описом.

Для контролю достовірності обчислень використовуються перевірки:

- перевірка валідності сесії перед кожною операцією;
- контроль унікальності `email` та `token` у базі даних;
- перевірка терміну дії посилань для спільного доступу, за формулою

2.3.4.

$$isValid = currentTime < expiresAt \quad (2.3.4)$$

Точність обчислень цін акцій забезпечується округленням до двох знаків після коми. Для уникнення помилок у базі даних застосовуються транзакції та зовнішні ключі (`FOREIGN KEY`).

Математичне забезпечення системи оптимізовано для швидкої обробки запитів та підтримки реального часу. Алгоритми побудовані з урахуванням всіх можливих сценаріїв (автентифікація, помилки, оновлення даних), а модель даних

є компактною, що зменшує обчислювальну складність. Використання SQLite та WebSocket забезпечує ефективність та масштабованість системи.

2.4. Рішення з програмного забезпечення

Розробка автоматизованого робочого місця інвестора IT-проєкту базується на комплексному підході до створення інформаційної системи, що забезпечує управління інвестиційними дошками, автентифікацію користувачів, спільне редагування та відстеження змін у реальному часі. Технічне забезпечення включає серверне та клієнтське програмне забезпечення, реалізоване на основі Java-технологій. Інформаційне забезпечення охоплює базу даних SQLite для зберігання даних про користувачів, дошки, акції та історію змін. Організаційні аспекти проєкту детально описані в технічному завданні проєкту.

Програмне забезпечення системи складається з двох основних компонентів: серверної частини (Server.java) та клієнтської частини (Client.java) (рисунки 2.3). Серверна частина відповідає за обробку запитів, управління базою даних, автентифікацію, WebSocket-з'єднання для реального часу та HTTP-сервер для вебінтерфейсу. Клієнтська частина реалізує графічний інтерфейс користувача (GUI) на основі Java Swing, забезпечуючи взаємодію з сервером через сокетне з'єднання. Взаємозв'язок між компонентами здійснюється через мережеві протоколи TCP для сокетів, WebSocket для оновлень у реальному часі та HTTP для вебдоступу. Виділення цих компонентів обґрунтовано необхідністю розподілу серверної логіки та клієнтського інтерфейсу для забезпечення масштабованості та зручності використання.

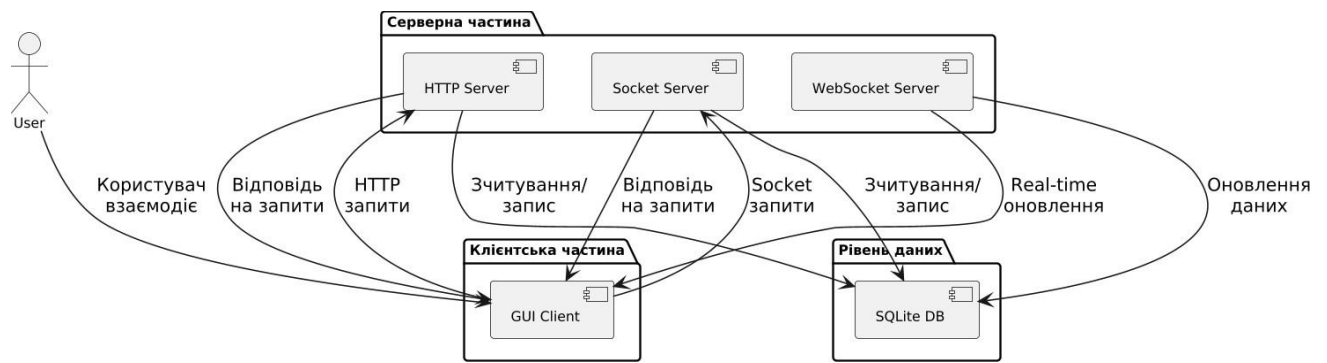


Рисунок 2.3 – Діаграма компонентів

Серверна частина виконує такі функції: автентифікація користувачів (включаючи 2FA та OAuth), управління інвестиційними дошками, обробка запитів на створення/редагування дошок, акцій і позначок, відстеження цін акцій через періодичні оновлення, запис історії змін і розсилання сповіщень через WebSocket. Клієнтська частина забезпечує відображення інтерфейсу для входу, реєстрації, перегляду списку дошок, їх деталізації (таблиці акцій, графічні позначки), а також інтерактивне додавання/редагування даних. Локалізація українською мовою підвищує доступність для користувачів.

Розробка проводилася з використанням об'єктно-орієнтованого програмування (ООП) на мові Java [15]. Для серверної частини застосовувалися бібліотеки Java: `java.net` для мережевих операцій, `java.sql` для роботи з SQLite, `com.sun.net.httpserver` для HTTP-сервера та `org.java_websocket` для WebSocket. Клієнтська частина використовує Java Swing для GUI. Інструменти розробки включали IDE (наприклад, IntelliJ IDEA), систему контролю версій Git і Maven для управління залежностями. Тестування проводилося шляхом модульного тестування та інтеграційного тестування клієнт-серверної взаємодії.

Система розроблена для роботи під управлінням операційних систем Windows, Linux або macOS, де підтримується Java Runtime Environment (JRE) версії 8 або вище. Вибір Java забезпечує кросплатформність, що дозволяє запускати систему без модифікацій на різних ОС. SQLite як легка вбудована база даних не потребує додаткових серверних компонентів, що спрощує розгортання.

Для реалізації проєкту використано Java SE для серверної та клієнтської логіки, SQLite для зберігання даних, бібліотеки `java.security` для шифрування паролів і TOTP для 2FA, а також `org.java_websocket` для реального часу. Ці засоби забезпечують надійність, безпеку та ефективну обробку даних. SQLite підтримує транзакції, що гарантує цілісність даних, а Java Swing забезпечує інтуїтивний інтерфейс.

Користувачам надаються такі сервісні можливості: управління дошками через вебінтерфейс (HTTP-сервер), сповіщення в реальному часі через WebSocket, підтримка електронної пошти для запрошень і сповіщень про цінові алерти. Система підтримує локалізацію українською мовою, що полегшує взаємодію для україномовних користувачів. Інтеграція з Google OAuth забезпечує альтернативний спосіб автентифікації.

Оцінка надійності системи спрямована на визначення її стабільності та безвідмовності. Оцінюються такі показники: імовірність безвідмовної роботи (P) і середній час напрацювання на відмову (MTBF). Враховуються фактори: надійність серверного обладнання, стабільність мережевих з'єднань, коректність обробки помилок. Обмеження: не враховуються апаратні збої клієнтських пристроїв.

Надійність елементів: Java-програми ($P=0,99$), SQLite ($P=0,98$), мережеві компоненти ($P=0,95$). Режим роботи: цілодобовий, з піковим навантаженням до 100 одночасних користувачів. Експлуатація: регулярне оновлення цін кожні 15 хвилин, очищення сесій кожні 30 хвилин.

Розрахунок проводився за методикою оцінки надійності складних систем, описаною в ДСТУ 27.002-2015. Надійність системи розраховується як добуток імовірностей безвідмовної роботи компонентів: $P_{сист} = P_1 \times P_2 \times \dots \times P_n$. MTBF оцінюється як середнє значення часу роботи між відмовами для кожного компонента.

Структура системи включає три основні компоненти: сервер (Java), база даних (SQLite), мережа. Імовірність безвідмовної роботи:

$$P_{сист} = 0,99 \times 0,98 \times 0,95 = 0,92169.$$

MTBF для Java-програм = 10000 год,

SQLite = 8000 год,

мережа = 5000 год.

Середнє *MTBF* системи: $(10000 + 8000 + 5000) / 3 \approx 7667$ год.

Імовірність безвідмовної роботи системи становить 0,9217, що є достатнім для інформаційних систем такого типу. *MTBF* (7667 год) свідчить про високу стабільність. Для підвищення надійності рекомендується впровадження резервного копіювання бази даних і моніторингу мережес'єднань.

2.5. Рішення з технічного та програмного забезпечення

Схема автоматизації розробленої інформаційної системи відображає об'єкт автоматизації – автоматизоване робоче місце інвестора ІТ-проєкту, яке забезпечує управління інвестиційними дошками, спільне редагування та відстеження змін у реальному часі. Технічне забезпечення включає серверну частину (Java-програма Server.java), клієнтську частину (Java-програма Client.java), базу даних SQLite та мережеву інфраструктуру. Сервер обробляє запити через сокетне з'єднання (порт 8080), HTTP (порт 8081) та WebSocket (порт 8082), забезпечуючи взаємодію з клієнтськими додатками. Функціональні зв'язки між компонентами реалізуються через сокетний протокол для клієнт-серверної взаємодії, HTTP для веб-інтерфейсу та WebSocket для оновлень у реальному часі. Зовнішні зв'язки передбачають інтеграцію з API фінансових даних (наприклад, Yahoo Finance) для оновлення цін акцій. Умовні позначки, не передбачені стандартами, включають позначення WebSocket-з'єднань (лінія з маркером "WS") та локальної бази даних (прямокутник із написом "SQLite"). Текстові пояснення на схемі уточнюють порти та протоколи обміну даними.

Технічне забезпечення системи розроблено з урахуванням вимог до високої доступності, безпеки та можливості масштабування. Вихідними даними для проєктування слугували вимоги до підтримки автентифікації (включаючи 2FA та OAuth), управління дошками, спільного редагування та моніторингу цін акцій.

Система розрахована на використання стандартного серверного обладнання та персональних комп'ютерів користувачів, що забезпечує економічну ефективність та універсальність.

Комплекс технічних засобів (КТЗ) складається з серверного обладнання, клієнтських пристроїв та мережевої інфраструктури. Серверна частина розгортається на апаратному сервері з операційною системою, що підтримує Java (наприклад, Linux/Windows), та включає базу даних SQLite для зберігання інформації про користувачів, дошки, акції та історію змін. Обмін даними з зовнішніми системами (фінансовими API) здійснюється через HTTPS-протокол, що забезпечує безпечну передачу даних. У пусковому режимі сервер ініціалізує базу даних, запускає HTTP та WebSocket-сервери, а також планує регулярні задачі (оновлення цін акцій, очищення сесій). В аварійних режимах передбачено автоматичне очищення прострочених сесій та логування помилок для діагностики. Розміщення КТЗ здійснюється в офісному середовищі з дотриманням вимог до електроживлення (стабільне джерело 220 В) та вентиляції для серверного обладнання. Безпека забезпечується хешуванням паролів (SHA-256), підтримкою 2FA (TOTP), захистом від SQL-ін'єкцій та CORS для веб-доступу. Надійність КТЗ оцінена як висока завдяки використанню стабільних бібліотек Java та локальної бази даних, що мінімізує залежність від зовнішніх сервісів.

Вибір персональних комп'ютерів для клієнтської частини базується на мінімальних вимогах: процесор рівня Intel Core i3, 4 ГБ оперативної пам'яті, ОС Windows/Linux/macOS з підтримкою Java SE 8+. Периферійні засоби включають монітори для відображення графічного інтерфейсу, клавіатуру та мишу для взаємодії з клієнтським додатком. Структурна схема обчислювальної мережі передбачає клієнт-серверну архітектуру з локальним сервером, підключеним до мережі Інтернет через маршрутизатор із пропускною здатністю не менше 100 Мбіт/с. Кількість технічних засобів залежить від числа користувачів: один сервер підтримує до 100 одночасних підключень, клієнтські ПК відповідають числу активних інвесторів (оціночно 10–50). Для зберігання даних використовується

локальна база SQLite, що не потребує додаткових носіїв. Обслуговування КТЗ забезпечується одним системним адміністратором для сервера та базовим навчанням користувачів для клієнтського ПЗ.

Розроблена інформаційна система відповідає вимогам до автоматизованого робочого місця інвестора, забезпечуючи надійну автентифікацію, управління інвестиційними дошками та реальний час оновлення даних. Технічне забезпечення є економічно ефективним, безпечним та масштабованим, що підтверджується використанням стандартного обладнання, стабільних протоколів та механізмів захисту даних. Система готова до впровадження з мінімальними вимогами до персоналу та інфраструктури.

Розроблено архітектуру системи, включно з рішеннями щодо інформаційного, математичного, програмного та технічного забезпечення, що дозволяє реалізувати функціональність АРМ інвестора з урахуванням безпеки, масштабованості та інтеграції з зовнішніми сервісами.

Розділ 3. Реалізація проєкту

3.1. Розробка робочої документації

Розробка робочої документації є ключовим етапом у процесі створення будь-якого проєкту, зокрема в галузі інформаційних технологій. Цей етап забезпечує чітке розуміння вимог, структури та функціональних особливостей системи для всіх учасників процесу — від розробників до замовників. У контексті реалізації проєкту детальна робоча документація слугує основою для подальшої імплементації, тестування та супроводу системи. Детальний розгляд етапу розробки робочої документації, розбитий на ключові пункти, що відображають його складові, принципи та підходи, описаний в табл. 3.1.

Таблиця 3.1 – Ключові пункти етапу розробки робочої документації

№	Пункти	Пояснення
1	2	3
1	Визначення цілей і завдань робочої документації	Першим кроком є формулювання мети створення документації. Вона полягає в забезпеченні повного опису системи, її компонентів і взаємодій між ними. Завдання включають створення інструкцій для розробників, підготовку матеріалів для тестувальників і формування бази знань для майбутнього супроводу. Важливо враховувати, що документація має бути зрозумілою не лише технічним спеціалістам, а й особам, які приймають рішення на рівні управління
2	Збір і аналіз вимог	На цьому етапі проводиться детальний аналіз потреб користувачів і замовників. Збираються функціональні та нефункціональні вимоги, які стають основою для технічних рішень. Наприклад, якщо проєкт передбачає розробку програмного забезпечення, необхідно визначити, які операції користувач зможе виконувати, а також вимоги до швидкодії чи безпеки. Цей процес потребує співпраці з усіма зацікавленими сторонами, щоб уникнути непорозумінь на подальших етапах
3	Структура документації	Створення робочої документації вимагає чіткої організації вмісту. Зазвичай вона включає такі розділи:

3.1	Огляд проєкту	короткий опис цілей і контексту
-----	---------------	---------------------------------

Продовження табл. 3.1

1	2	3
3.2	Технічна специфікація	деталі реалізації, включаючи архітектуру системи
3.3	Опис компонентів	характеристика модулів, їх функцій і взаємозв'язків
3.4	Інструкції з розгортання	кроки для запуску системи в робочому середовищі
3.5	План тестування	критерії перевірки працездатності. Кожен із цих елементів розробляється з урахуванням специфіки проєкту та потреб аудиторії
4	Вибір інструментів для документування	Для підготовки матеріалів використовуються різноманітні засоби, такі як текстові редактори (наприклад, Microsoft Word), системи управління версіями (Git) або спеціалізоване програмне забезпечення (Confluence, Sphinx). Вибір залежить від складності проєкту та звичок команди. Наприклад, для технічних команд, які працюють із кодом, доцільно інтегрувати документацію в репозиторії, щоб полегшити доступ і оновлення
5	Опис архітектури системи	Одним із найважливіших елементів є детальний виклад архітектури. Це передбачає створення схем, які ілюструють зв'язки між модулями, потоки даних і принципи роботи системи. Наприклад, якщо проєкт базується на клієнт-серверній моделі, необхідно описати, як клієнтська частина взаємодіє із сервером, які протоколи застосовуються (HTTP, WebSocket тощо) і як обробляються запити
6	Деталізація модулів і функцій	Кожен компонент системи потребує окремого опису. Для цього зазначаються його призначення, вхідні та вихідні дані, а також логіка роботи. Наприклад, якщо в проєкті є модуль автентифікації, документація має пояснити, як користувач проходить перевірку, які алгоритми шифрування застосовуються (наприклад, bcrypt) і як обробляються помилки, такі як неправильний пароль
7	Врахування стандартів і нормативів	Робоча документація повинна відповідати прийнятим у галузі стандартам, наприклад, ISO/IEC 12207, що регулює процеси життєвого циклу програмного забезпечення. Дотримання таких норм забезпечує якість і можливість подальшої сертифікації продукту, якщо це необхідно. Також важливо враховувати внутрішні стандарти компанії чи команди, щоб усі документи мали однаковий формат і стиль

Завершення табл. 3.1

1	2	3
8	Ілюстративні матеріали	Для кращого сприйняття текст доповнюється діаграмами (UML, ERD), графіками й таблицями. Наприклад, діаграма класів може показати ієрархію об'єктів у коді, а схема бази даних — структуру таблиць і зв'язків між ними. Такі візуальні елементи значно полегшують розуміння складних технічних деталей
9	Процес узгодження	Після створення чернетки документація проходить етап рецензування. У ньому беруть участь розробники, аналітики, тестувальники та замовники. Метою є виявлення неточностей, уточнення деталей і забезпечення відповідності матеріалів реальним потребам проєкту. Цей етап може включати кілька ітерацій, доки не буде досягнуто консенсусу
10	Оновлення та підтримка	Робоча документація не є статичним продуктом. У процесі реалізації проєкту можуть виникати зміни в вимогах чи технічних рішеннях, що потребує внесення коректив. Для цього доцільно вести журнал змін, який фіксує дату, суть оновлення та відповідальну особу. Такий підхід дозволяє уникнути плутанини та зберегти актуальність матеріалів
11	Передача документації команді	Після завершення підготовки документи передаються учасникам реалізації проєкту. Важливо забезпечити легкий доступ до них, наприклад, через хмарні сховища чи внутрішні портали. Крім того, варто провести короткий інструктаж, щоб пояснити, як користуватися матеріалами та де шукати потрібну інформацію
12	Оцінка якості документації	На завершальному етапі розробки проводиться аналіз її повноти та зрозумілості. Для цього можна залучити зовнішніх експертів або використати контрольні списки, які включають питання на кшталт: “Чи достатньо детально описано функції?”, “Чи зрозумілі інструкції новачку?”. Така оцінка допомагає вдосконалити матеріали перед початком активної фази реалізації

Розробка робочої документації є невід'ємною частиною реалізації проєкту, що забезпечує його успішне втілення в життя. Цей процес вимагає ретельного планування, співпраці між учасниками та уваги до деталей. Завдяки чітко структурованій і деталізованій документації команда отримує надійний орієнтир для роботи, а замовник — прозоре уявлення про хід виконання завдань. Усе це

разом сприяє досягненню поставлених цілей і створенню якісного продукту, готового до використання та подальшого розвитку.

3.2. Екранні форми інтерфейсу користувача

Розробка програмного забезпечення потребує ретельного створення зручного та функціонального інтерфейсу користувача. Екранні форми є ключовим елементом взаємодії між користувачем і системою, визначаючи зручність роботи, доступність даних і ефективність виконання завдань. У процесі реалізації проєкту акцент зроблено на продуманому дизайні екранних форм, які відповідають потребам користувачів, забезпечують логічну структуру та інтуїтивну навігацію. Особливості створення екранних форм, їх проектування, функціональність і практична цінність для продукту детально розглянуто в табл. 3.2.

Таблиця 3.2 – Особливості створення екранних форм інтерфейсу користувача

№	Особливості	Пояснення
1	2	3
1	Визначення структури екранних форм	Початковий етап передбачає аналіз потреб користувачів і функціоналу системи. Створюється структура з елементами інтерфейсу: кнопками (наприклад, "Увійти" чи "Створити дошку"), текстовими полями (email, пароль), списками (дошки користувача). Логічне розміщення елементів уникає перевантаження. Структура адаптується до сценаріїв використання, як-от перемикання між панелями входу та дошок у Client.java.
2	Принципи дизайну інтерфейсу	Дизайн базується на ергономії та візуальній чіткості. Використовуються зрозумілі шрифти й кольори (наприклад, червоний для позначок типу "ARROW" у createMarkerLabel). Розташування елементів відповідає руху очей: зліва направо, зверху вниз, як у GridBagLayout для панелей входу та реєстрації, що полегшує освоєння інтерфейсу.

Продовження табл. 3.2

1	2	3
3	Функціональність та інтерактивність	Форми забезпечують активну взаємодію: спливні підказки (tooltips для позначок), динамічні списки (оновлення дощок у boardsListModel). У Client.java кнопки викликають дії (login(), addStock()), зменшуючи помилки завдяки зворотному зв'язку від сервера.
4	Адаптивність до різних пристроїв	Форми розроблено з урахуванням різних розмірів екранів. У Client.java використано JScrollPane для прокручування таблиць і панелей, що дозволяє вертикальне розміщення на малих екранах. Розміри компонентів (наприклад, 800x600) адаптуються до роздільної здатності.
5	Валідація даних	Реалізовано перевірку введених даних. У Client.java порожні поля email і пароля викликають повідомлення ("Введіть email та пароль"), а числові значення для акцій (quantity, buyPrice) перевіряються через try-catch. Сервер у Server.java також валідажує параметри (наприклад, довжину пароля).
6	Тестування та оптимізація	Після розробки форми тестуються для виявлення недоліків. У Client.java обробка винятків (IOException, NumberFormatException) із виведенням повідомлень користувачу дозволяє коригувати інтерфейс. Оптимізація швидкості досягається через асинхронні запити до сервера (handleClient).
7	Локалізація та персоналізація	Форми адаптовано до української мови через LOCALIZATION у Client.java (наприклад, "Email:", "Додати акцію"). Формати дат відповідають локальним стандартам (LocalDateTime). Користувач може налаштувати дошки чи позначки, зберігаючи власні дані.
8	Забезпечення доступності	Інтерфейс враховує потреби різних користувачів. У Client.java контрастні кольори позначок (червоний, синій) полегшують сприйняття. Підтримка клавіатурного введення та сповіщень (JOptionPane) робить систему зручною для осіб із порушеннями зору.
9	Інтеграція з іншими компонентами системи	Форми пов'язані з сервером (Server.java) через сокети та HTTP-запити. Дані з форм (addStock) передаються для збереження в базі SQLite, а відповіді (SUCCESS

Розробка екранних форм інтерфейсу користувача – складний процес, що вимагає уваги до деталей і орієнтації на користувача. Від логічного розміщення

кнопок до адаптивності та доступності кожен етап впливає на якість продукту. Успішна реалізація підвищує ефективність системи, комфорт користувачів і сприяє досягненню цілей проєкту, відповідаючи сучасним стандартам і потребам аудиторії.

Для візуалізації екранних форма варто розглянути покрокове виконання програми "Автоматизоване робоче місце інвестора ІТ-проєкту"

Крок 1. Запуск програми

Програма розпочинає роботу з ініціалізації серверної частини. Спочатку запускається сервер, який виводить повідомлення про початок роботи та намагається підключитися до бази даних SQLite. У цей момент також створюються необхідні таблиці для зберігання інформації про користувачів, дошки, акції, позначки тощо. Одночасно клієнтська частина відкриває графічний інтерфейс із панеллю входу (рисунок 3.1), намагаючись встановити з'єднання з сервером через сокет.

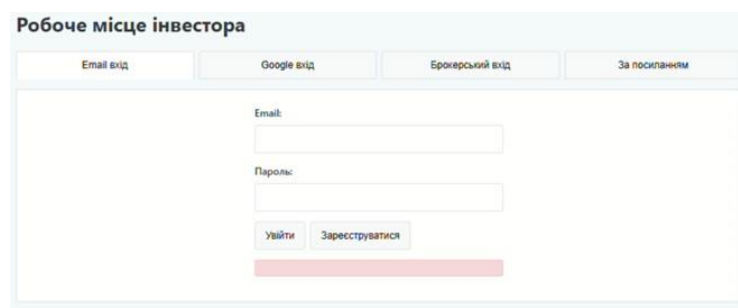


Рисунок 3.1 – Панель входу

Крок 2. Ініціалізація серверних компонентів

Сервер налаштовує регулярні завдання: очищення прострочених сесій, оновлення цін акцій та видалення неактивних посилань для доступу. Запускаються HTTP-сервер для веб-інтерфейсу та WebSocket-сервер для оновлень у реальному часі. Після цього сервер переходить у режим очікування клієнтських підключень через сокети.

Крок 3. Вхід користувача

Користувач вводить email та пароль у відповідні поля на панелі входу клієнтського інтерфейсу. Натискаючи кнопку "Увійти" (рисунок 3.2), клієнт надсилає запит до сервера. Сервер перевіряє облікові дані в базі даних, і в разі успіху створює сесію, повертаючи її ідентифікатор клієнту. Якщо увімкнена двофакторна автентифікація, користувачу пропонується ввести код підтвердження (рисунок 3.3).

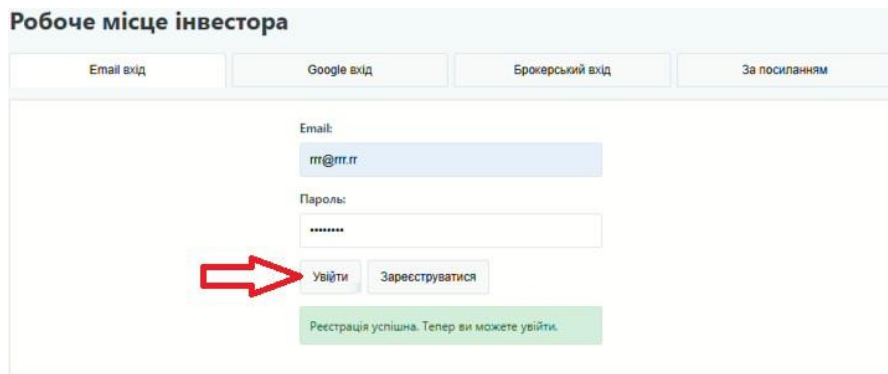


Рисунок 3.2 – Вхід користувача до системи

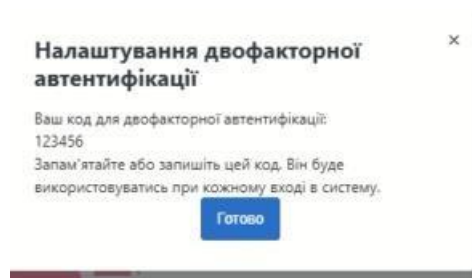


Рисунок 3.3 – Налаштування двофакторної автентифікації

Кроки 4. Реєстрація нового користувача

Якщо користувач обирає "Реєстрація" (рисунок 3.4, рисунок 3.5), відкривається відповідна панель. Після введення email та пароля дані надсилаються серверу, який перевіряє унікальність email. У разі успіху створюється новий запис у базі даних, і користувач отримує повідомлення про можливість входу.

The screenshot shows a registration form with four tabs at the top: "Email вхід", "Google вхід", "Брокерський вхід", and "За посиланням". The "Email вхід" tab is selected. Below the tabs, there are two input fields: "Email:" with the value "m@m.m" and "Пароль:" with masked characters "*****". Below the password field are two buttons: "Увійти" and "Зареєструватися". At the bottom, a red error message states: "Пароль повинен містити не менше 8 символів".

Рисунок 3.4 – Помилка при реєстрації

The screenshot shows the same registration form as in Figure 3.4, but with a green success message at the bottom: "Реєстрація успішна. Тепер ви можете увійти." The "Зареєструватися" button is now disabled, and the "Увійти" button is highlighted.

Рисунок 3.5 – Успіх реєстрації

Крок 5. Завантаження списку дощок

Після успішного входу клієнт переходить на панель із переліком дощок (рисунок 3.6). Сервер повертає список доступних дощок, пов'язаних із користувачем, які відображаються у вигляді списку. Користувач може обрати дошку для детального перегляду або створити нову.

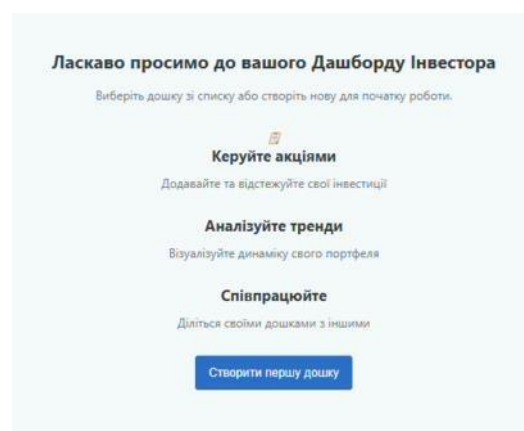


Рисунок 3.6 – Перелік дощок

Крок 6. Створення дошки

Користувач натискає кнопку "Створити дошку", вводить її назву у діалоговому вікні (рисунок 3.7). Клієнт надсилає запит серверу, який додає дошку до бази даних, призначає власника та записує подію в історію змін. Список дощок оновлюється автоматично.

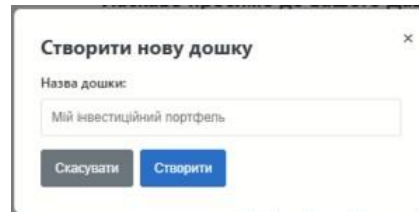


Рисунок 3.7 – Створення нової дошки

Крок 7. Запрошення користувача до дошки

Обрана дошка дозволяє запросити іншого користувача через кнопку "Запросити" (рисунок 3.8). Вводиться email та обираються права доступу. Сервер перевіряє права власника, додає запрошеного користувача до дошки та оновлює історію.

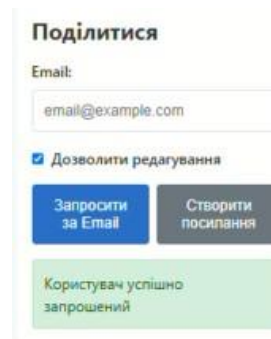


Рисунок 3.8 – Запрошення користувача

Крок 8. Додавання акцій

На панелі деталей дошки користувач натискає "Додати акцію" (рисунок 3.9), вводить символ, кількість, ціну купівлі та примітки. Сервер зберігає дані в базі даних, оновлює таблицю акцій і надсилає повідомлення через WebSocket усім підписаним клієнтам (рисунок 3.10).

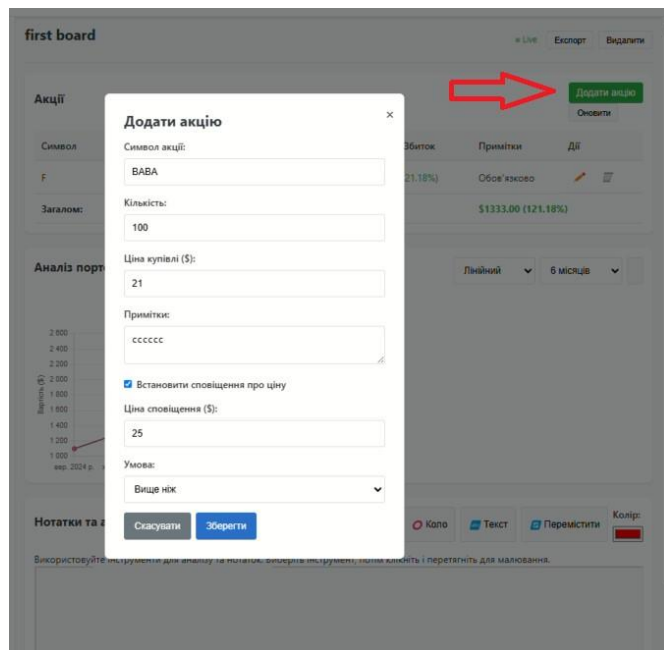


Рисунок 3.9 – Додавання акцій

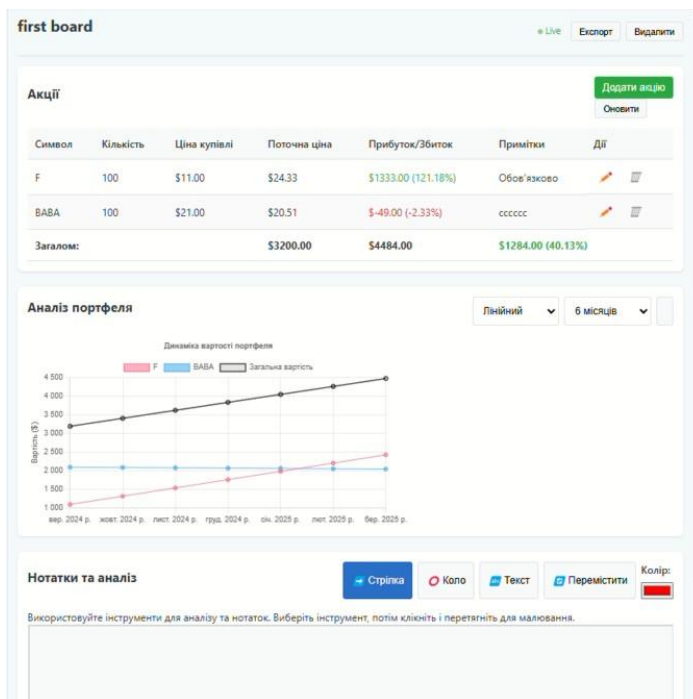


Рисунок 3.10 – Оновлення даних дошки після додавання акцій

Крок 9. Робота з позначками

Користувач додає графічні позначки (стрілки, кола, текст) на дошку через кнопку "Додати позначку" (рисунок 3.11). Вказуються тип, координати та опис. Сервер записує позначку, оновлює дошку та розсилає зміни підписникам.

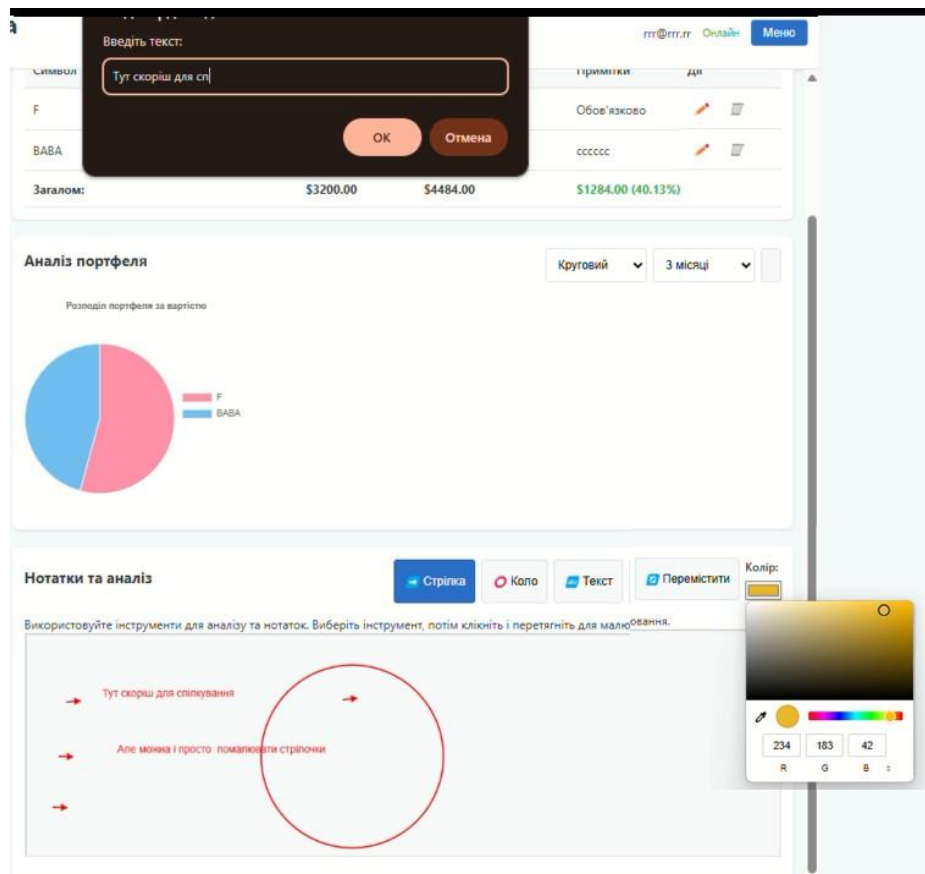


Рисунок 3.11 – Робота з позначками

Крок 10. Оновлення цін акцій

Сервер періодично (кожні 15 хвилин) оновлює ціни акцій із зовнішнього джерела (симуляція), зберігає їх у базі даних та перевіряє умови сповіщень. Якщо ціна досягає заданого рівня, надсилається повідомлення через WebSocket.

Крок 11. Перегляд деталей дошки

Обрана дошка відображає таблицю акцій та графічні позначки (рисунок 3.12). Клієнт запитує деталі в сервера, який повертає актуальні дані, включаючи історію змін за потреби.

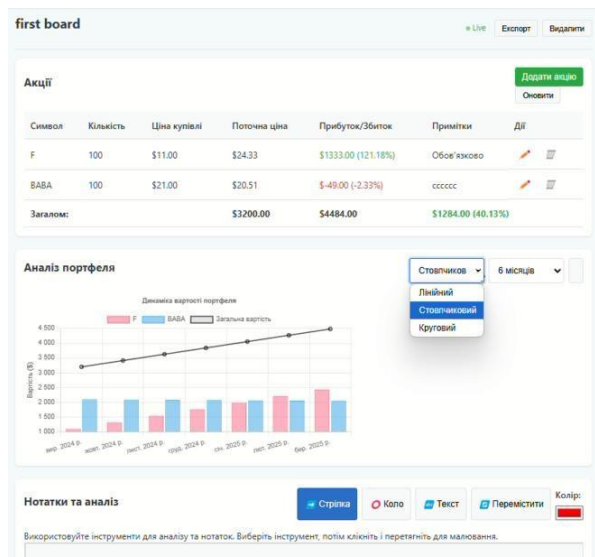


Рисунок 3.12 – Перегляд деталей дошки

Крок 12. Вихід із системи

Користувач завершує роботу, натиснувши "Вихід" (рисунок 3.13). Клієнт надсилає запит на видалення сесії, сервер очищає її з бази даних і пам'яті, повертаючи користувача до панелі входу.

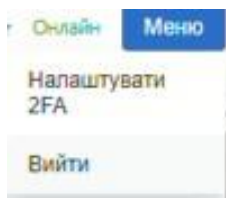


Рисунок 3.13 – Вихід із системи

Крок 13. Завершення роботи програми

Закриття клієнтського вікна розриває з'єднання з сервером. Сервер продовжує працювати, обробляючи інших клієнтів, доки не буде зупинений адміністратором. Усі ресурси, такі як сокети та потоки, коректно закриваються.

Цей процес забезпечує повноцінну взаємодію між клієнтом і сервером, підтримуючи функціональність автоматизованого робочого місця інвестора.

Система була успішно реалізована як клієнт-серверна платформа з інтерфейсами для користувача та адміністратора, забезпечено збереження й

актуалізацію даних, протестовано основні функції, включаючи автентифікацію, обмін інформацією, управління портфелем і співпрацю користувачів.

Розділ 4. Охорона праці при роботі за комп'ютером

Сучасний технічний та технологічний розвиток виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. На даний час важко уявити підприємство на якому відсутнє використання комп'ютерної техніки.

Робота з комп'ютерною технікою пов'язана з низкою факторів ризиків, які можуть негативно впливати на здоров'я працівників, а саме це навантаження на зір, навантаження на опорно-руховий апарат, психоемоційне напруження, вплив електромагнітного випромінювання, шуму та недостатнього освітлення. З метою створення безпечних умов праці роботодавець повинен дотримуватись вимог чинного законодавства України.

Відповідно до статті 13 Закону України «Про охорону праці» [16], роботодавець повинен створити на робочому місці умови праці відповідно до нормативно – правових актів, а також повинен забезпечити дотримання прав працівників у галузі охорони праці.

З метою створення комфортних, безпечних та здорових умов праці при роботі за комп'ютером роботодавець повинен керуватися такими нормативно – правовими актами:

- наказом Мінсоцполітики «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» від 14.02.2018 р. № 207 [17], який поширюється на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми і видів діяльності та встановлює мінімальні вимоги безпеки та захисту здоров'я під час здійснення роботи, пов'язаної з використанням екранних пристроїв незалежно від їхнього типу та моделі;
- Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затвердженими постановою Головного державного санітарного лікаря України 10.12.1998 р. № 7, що застосовуються на підприємствах, організаціях, установах незалежно від

форми власності та поширюються на умови й організацію праці при роботі з візуальними дисплейними терміналами усіх типів вітчизняного та закордонного виробництва на основі електронно-променевих трубок, якими укомплектовані електронно-обчислювальні машини колективного використання та персональні комп'ютери.

Відповідно до п. 5 наказу № 207, роботодавець повинен за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку працівників. Ці перерви встановлюються згідно з вимогами ДСанПіН 3.3.2.007-98 затверджених постановою Головного державного санітарного лікаря України від 10 грудня 1998 року № 7.

Розробники програмного забезпечення: 15 хвилин відпочинку після кожної години роботи за комп'ютером. Оператори ПК: 15 хвилин після кожних двох годин роботи. Оператори комп'ютерного набору: 10 хвилин після кожної години роботи.

У разі неможливості запровадження перерв згідно з графіком, безперервна робота за комп'ютером не повинна перевищувати 4 годин. При 12-годинній робочій зміні, перерви організовуються в перші 8 годин за стандартною схемою, а в останні 4 години — через кожну годину по 15 хвилин, незалежно від характеру роботи.

Роботодавець зобов'язаний забезпечити комфортні та безпечні умови праці на кожному робочому місці. Мінімальна площа робочого місця має становити не менше 6 метрів квадратних. Якщо поруч працюють кілька працівників за комп'ютерами, їх робочі зони за потреби слід розмежовувати перегородками висотою до 2 метрів.

Під час визначення достатньої площі приміщення слід враховувати розміри всіх предметів меблів та обладнання (шаф, тумб, сейфів тощо), що знаходяться в кімнаті.

Основними елементами робочого місця є персональний комп'ютер або ноутбук, монітор, клавіатура, робочий стіл і крісло. Усі складові повинні бути

розміщені так, щоб сприяти ефективному та зручному виконанню трудових завдань.

Допускається розміщення на столі додаткових пристроїв (принтера, сканера, колонок) та засобів для зберігання документів, за умови, що вони не перекривають огляд екрана і не заважають працівнику.

У випадку підвищеного рівня шуму чи вібрації від техніки, працівникам слід надати антивібраційні килимки. Комп'ютери необхідно встановлювати на відстані не менше 1 метра від джерел тепла.

Не допускається поява світлових відблисків на екрані, клавіатурі чи інших частинах комп'ютера, які можуть потрапляти в очі користувача.

Рекомендується розташовувати комп'ютери в приміщенні так, щоб природне освітлення надходило на екрани моніторів зі сторони півдня або північного сходу.

Монітор слід встановлювати на відстані 600–700 мм від очей користувача, що залежить від його діагоналі. Найкраще, щоб екран був розміщений перпендикулярно до напрямку погляду користувача.

Також екран варто орієнтувати так, щоб світло з вікна падало збоку, бажано зліва, не спричиняючи відблисків.

Клавіатуру розміщують на робочому або спеціальному столі на відстані 100–300 мм від переднього краю. Її нахил встановлюється за бажанням користувача, зазвичай у межах 5–15°. При цьому клавіатура має бути стійкою і не хитатися.

Висота робочої поверхні стола має бути регульованою в межах 680–800 мм, оптимальне значення — близько 725 мм.

Крісло працівника повинно мати механізми підйому і повороту, можливість регулювання по висоті, а також забезпечувати зручну підтримку спини та хребта.

Електрична розетка повинна бути розташована у легкодоступному місці, щоб у разі аварії можна було швидко вимкнути живлення.

У приміщеннях, де використовуються комп'ютери, слід щоденно проводити вологе прибирання та регулярно провітрювати кімнату протягом дня. Пил з екрана необхідно видаляти щонайменше один раз на добу.

Дотримання нормативно-правових вимог щодо організації праці з екранними пристроями є обов'язковою умовою збереження здоров'я працівників. Забезпечення належного режиму праці та відпочинку, організація комфортного та безпечного робочого місця дозволяє мінімізувати ризики професійних захворювань і підвищити ефективність праці.

Висновки

У даній кваліфікаційній роботі було розглянуто та вирішено актуальну задачу розробки автоматизованого робочого місця інвестора ІТ проєкту. Проведене дослідження дозволило виявити ключові потреби інвесторів у сучасних умовах та сформувані вимоги до функціоналу такого програмного забезпечення.

Зокрема, в ході виконання роботи було:

1. Проаналізовано основні інновації та технології, що впливають на процеси прийняття рішень. Розглянуто існуючі підходи до підтримки інвестиційних рішень та визначено їх переваги та недоліки.
2. Визначено ключові вимоги до АРМ інвестора ІТ проєкту.
3. Спроектовано архітектуру програмного комплексу АРМ інвестора. Вибір технологій розробки обґрунтовано з точки зору їх ефективності та відповідності сучасним стандартам.
4. Розроблено та реалізовано основні функціональні модулі АРМ, що включають систему управління проєктами, аналітичні інструменти для оцінки ризиків та прибутковості, а також модулі для взаємодії з фінансовими даними.
5. Проведено тестування розробленого АРМ, що підтвердило його відповідність заявленим вимогам та ефективність у вирішенні поставлених задач.
6. Розглянуті питання охорони праці при роботі за комп'ютером.

Таким чином, автоматизоване робоче місце інвестора ІТ проєкту є інноваційним та практично цінним рішенням, яке має потенціал суттєво покращити ефективність інвестиційних процесів у сфері ІТ. Воно надає інвесторам комплексний інструментарій для раціонального управління активами, зниження ризиків та максимізації прибутку.

Подальші дослідження та розробки можуть бути спрямовані на інтеграцію розширених функцій штучного інтелекту та машинного навчання для перспективного аналізу ринкових тенденцій, виявлення прихованих закономірностей у даних, автоматичної генерації інвестиційних рекомендацій та адаптацій до індивідуальних інвестиційних стратегій користувача. Також перспективним є розширення функціоналу для підтримки колаборативної роботи кількох інвесторів або команд, а також інтеграція з блокчейн-технологіями для підвищення прозорості та безпеки інвестиційних транзакцій.

Список використаних джерел

1. Федішин І.Б. Управління проектами в підприємницькій діяльності (опорний конспект лекцій для студентів спеціальності 7.03060101 «Менеджмент підприємницької діяльності» усіх форм навчання) / І.Б. Федішин. – Тернопіль, ТНТУ імені Івана Пулюя, 2016. – 161 с.
2. Інвестиційні можливості в українському ІТ-секторі: Роль інституційних інвесторів [Електронний ресурс] - Режим доступу: <https://www.assets.com.ua/posts-afg/investiciyni-mozhливosti-v-ukrayinskomu-it-sektori-rol-instituciynih-investoriv>
3. Інвестування: основні терміни та поняття_[Електронний ресурс] - Режим доступу: <https://drukarnia.com.ua/articles/investuvannya-osnovni-termini-ta-ponyattya-KuuMr>
4. Ляхович Л.А., МОНІТОРИНГ ІНВЕСТИЦІЙНИХ ПРОЕКТІВ ПІДПРИЄМСТВ. Всеукраїнський науково-виробничий журнал ІННОВАЦІЙНА ЕКОНОМІКА 5'2012[31]
5. AngelList__поняття__[Електронний ресурс] - Режим доступу: <https://www.angellist.com/>
6. Carta [Електронний ресурс] - Режим доступу: <https://carta.com/uk/en/>
7. Майбутнє управління проектами: як технології змінюють галузь [Електронний ресурс] - Режим доступу: <https://nofluffjobs.com/uk/log/robota-v-it/majbutnie-upravlinnia-proektami/>
8. Judith Nwoke. Digital Transformation in Financial Services and FinTech: Trends, Innovations and Emerging Technologies. International Journal of Finance 9(6):1-24. 2024
9. Золялетдінов К. АНАЛІЗ СУЧАСНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ: АЛГОРИТМИ АВТОМАТИЗОВАНОГО ПІДТРИМКИ. Наукові нотатки. 2022

10. Шведун В. О., Ахмедова О. О. СТРАТЕГІЧНІ ОРІЄНТИРИ РОЗВИТКУ ДЕРЖАВНОГО УПРАВЛІННЯ КОНКУРЕНТОСПРОМОЖНІСТЮ РЕГІОНІВ УКРАЇНИ. ДЕРЖАВНЕ УПРАВЛІННЯ: УДОСКОНАЛЕННЯ ТА РОЗВИТОК № 2 (2025)

11. OWASP Top 10 – 2021 [Електронний ресурс] - Режим доступу: <https://owasp.org/Top10/>

12. Підтримка проєктів зі значними інвестиціями: гайд для інвесторів [Електронний ресурс] - Режим доступу: <https://me.gov.ua/News/Detail/3ffa34a9-ae2d-49d6-9949-c4987f05fd25>

13. Що таке SQLite, для чого і як його використовувати [Електронний ресурс] - Режим доступу: <https://foxminded.ua/prohramuvannia-baz-danykh-na-sqlite/>

14. SQLite Типи даних із прикладами: Int, Text, Numeric, Real, Blob [Електронний ресурс] - Режим доступу: <https://www.guru99.com/uk/sqlite-data-types.html>

15. Що таке Java і де вона використовується [Електронний ресурс] - Режим доступу: <https://goit.global/ua/articles/shcho-take-java-i-de-vona-vykorystovuietsia/>

16. Закон України про охорону праці [Електронний ресурс] - Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

17. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс] - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

Додатки

Додаток А

Інструкція користувача

Назва системи: Автоматизоване робоче місце інвестора ІТ-проєкту
Призначення: Надання інвесторам інструментів для керування інвестиційними портфелями, спільної роботи та аналізу ринкових даних.

Основні кроки користування:

1. Запуск системи:
 - Запустіть `investor-platform-1.0-SNAPSHOT.jar` (або `.jar-with-dependencies.jar`) з папки `target`.
 - Клієнтський інтерфейс відкривається через `Client.java` (або веб через `index.html`).
2. Авторизація:
 - Використовуйте email та пароль.
 - Доступна 2FA-автентифікація та вхід через Google (OAuth).
3. Управління портфелем:
 - Створіть "дошку" (портфель).
 - Додавайте акції (символ, кількість, ціна).
 - Встановлюйте позначки, примітки, кольори.
4. Спільна робота:
 - Надішліть посилання на дошку іншим користувачам (можна з правами лише для перегляду).
 - Редагування доступне в реальному часі (WebSocket).
5. Сповіщення та аналіз:
 - Система надсилає сповіщення при досягненні встановлених цін.
 - Дані оновлюються кожні 15 хв.

Фрагменти лістингу

Client.java (GUI інтерфейс):

```
JFrame frame = new JFrame("Investor Dashboard");
JTable stockTable = new JTable(new DefaultTableModel());
frame.add(new JScrollPane(stockTable), BorderLayout.CENTER);
frame.setSize(800, 600);
frame.setVisible(true);
```

Server.java (робота з WebSocket, безпека):

```
ServerSocket serverSocket = new ServerSocket(8080);
while (true) {
    Socket client = serverSocket.accept();
    new Thread(new ClientHandler(client)).start();
}
```

База даних (SQLite):

```
CREATE TABLE users (
    id INTEGER PRIMARY KEY,
    email TEXT UNIQUE,
    password TEXT,
    use_2fa INTEGER,
    totp_secret TEXT
);
```


Додаток В**Програма та методика випробувань (ПМВ)****Мета:**

Перевірити коректність роботи АРМ інвестора.

Методика:

№	Тест	Вхідні дані	Очікуваний результат	Статус
1	Авторизація користувача	email + пароль	Вхід успішний	✓
2	Двофакторна автентифікація	email + TOTP	Вхід успішний	✓
3	Створення дошки	Назва, фон	Дошка створена	✓
4	Додавання акцій	AAPL, 10 шт	Акція відображається	✓
5	Візуалізація позначок	Текст, колір	Елемент з'являється	✓
6	Обмін дошкою	Сторонній email	Спільний доступ надано	✓
7	Сповіщення	AAPL > \$200	Надходить повідомлення	✓