

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему:

«Інформаційна система з адаптивним інтерфейсом для осіб з інклюзією»
здобувачки Козачук Дарини Романівни

Науковий керівник:

к.е.н., доцент

_____ Ріппа С.П.

**Робота допущена до захисту перед
екзаменаційною комісією з**

атестації здобувачів вищої освіти

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці
ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:
Керівник проєктної групи(гарант)
освітньо-професійної програми
_____ Помазун О.М.
“ _____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:
Завідувач кафедри
_____ Тішков Б.О.
“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачки вищої освіти *Козачук Дарини Романівни*

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему:

«Інформаційна система з адаптивним інтерфейсом для осіб з інклюзією»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р. № 466- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах
навчальних посібників та матеріалів з мережі Інтернет.

План кваліфікаційної бакалаврської роботи

Розділ I. Характеристика та аналіз інформаційних систем для користувачів з інвалідністю.

Розділ II. Розробка вимог та моделювання інформаційної системи для користувачів з інвалідністю.

Розділ III. Проектування і реалізація прототипу інформаційної системи.

Об'єктом дослідження є процеси розробки веб-орієнтованих інформаційних систем, спрямованих на забезпечення доступу до цифрових сервісів для користувачів з інклюзією.

Предметом дослідження є методи, інструменти та підходи до проектування інтерфейсів, що адаптуються до індивідуальних потреб користувачів з різними

формами інвалідності (порушеннями зору, моторики, когнітивного сприйняття).

Мета кваліфікаційної бакалаврської роботи є розробка концепції інформаційної системи з адаптивним інтерфейсом, яка забезпечує зручну та доступну взаємодію для осіб з інклюзією через персоналізовані UI-рішення та інтерактивний прототип.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I: Охарактеризувати проблематику цифрової інклюзії та інформаційних систем, орієнтованих на осіб з інклюзією, проаналізувати літературні джерела і практичний досвід використання адаптивних ІС, які мають потенціал використання в інклюзивних технологіях.

У розділі II: Сформулювати специфікації вимог, бізнес-вимоги і функціональні вимоги до ІС для осіб з інклюзією, виконати постановку задач, алгоритмізацію і моделювання інклюзивних ІС.

У розділі III: Виконати проєктування і реалізувати прототип інформаційної системи, орієнтованої на осіб з інклюзією.

**Завдання підготував
науковий керівник**

_____ **Ріппа Сергій Петрович**
« 10 » березня 2025 р.

**Завдання одержала
здобувачка**

_____ **Козачук Дарина Романівна**
« 10 » березня 2025 р.

Відгук
на кваліфікаційну магістерську роботу
здобувачки *Козачук Дарини Романівни* на тему:
«ІНФОРМАЦІЙНА СИСТЕМА З АДАПТИВНИМ
ІНТЕРФЕЙСОМ ДЛЯ ОСІБ З ІНКЛЮЗІЄЮ»
за спеціальністю 122 «Комп'ютерні науки»

Актуальність теми дослідження. Обрана здобувачем тематика кваліфікаційної бакалаврської роботи є безперечно актуальною, оскільки присвячена дослідженню та практичному впровадженню сучасних інформаційних технологій у сфері інклюзивного доступу до цифрових ресурсів. У контексті цифрової трансформації суспільства та зростання уваги до потреб осіб з інвалідністю, створення адаптивних інтерфейсів є важливою складовою забезпечення рівного доступу до інформації та сервісів.

Наукова новизна. Представлена до захисту робота містить елементи наукової новизни, оскільки автор розвиває концепцію створення інформаційних систем, орієнтованих на користувачів з різними типами порушень. У роботі проаналізовано сучасні підходи до адаптивного дизайну, запропоновано власну архітектуру системи з урахуванням принципів доступності, а також реалізовано прототип, що демонструє можливості персоналізації інтерфейсу відповідно до потреб користувача.

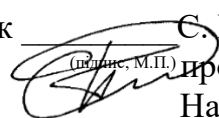
Якість проведеного аналізу проблеми, текстового та табличного матеріалу. Кваліфікаційна бакалаврська робота містить ґрунтовний аналіз існуючих рішень у сфері інклюзивних ІТ-систем, обґрунтування вибору технологій та підходів до реалізації адаптивного інтерфейсу. У роботі представлено структуровані діаграми, моделі даних, приклади інтерфейсів, що ілюструють логіку функціонування системи. Матеріал викладено послідовно, з дотриманням вимог до оформлення.

Обґрунтованість пропозицій і висновків та їх практична цінність. Підготовлені в роботі пропозиції щодо реалізації адаптивного інтерфейсу є обґрунтованими, а висновки — логічно узагальнюють результати дослідження. Розроблена система має прикладне значення та може бути використана в освітніх установах, громадських організаціях або державних сервісах, що працюють з людьми з інклюзією. Вона також має потенціал для подальшого розвитку та масштабування.

Наявність недоліків. Суттєвих зауважень до виконаної роботи не має. Окремі стилістичні неточності не впливають на загальну якість дослідження.

Загальний висновок і оцінка роботи. Кваліфікаційна бакалаврська робота на тему «ІНФОРМАЦІЙНА СИСТЕМА З АДАПТИВНИМ ІНТЕРФЕЙСОМ ДЛЯ ОСІБ З ІНКЛЮЗІЄЮ» у цілому відповідає встановленим вимогам вищої школи та рекомендується до захисту, а її авторка *Козачук Дарина Романівна* заслуговує на присвоєння освітньо-кваліфікаційного рівня «бакалавр» за спеціальністю 122 «Комп'ютерні науки».

Науковий керівник



С. П. Ріппа, д.е.н., професор,

професор кафедри інформаційних систем в економіці
Навчально-наукового інституту "Інститут
інформаційних технологій в економіці" КНЕУ

РЕЦЕНЗІЯ
на кваліфікаційну бакалаврську роботу
здобувачки *Козачук Дарини Романівни* на тему
«ІНФОРМАЦІЙНА СИСТЕМА З АДАПТИВНИМ ІНТЕРФЕЙСОМ ДЛЯ
ОСІБ З ІНКЛЮЗІЄЮ»
за спеціальністю 122 «Комп'ютерні науки»

Актуальність теми. Представлена на захист кваліфікаційна бакалаврська робота присвячена надзвичайно актуальній тематиці, що стосується розробки інформаційних систем, орієнтованих на потреби осіб з інклюзією. У сучасних умовах цифрової трансформації суспільства питання доступності цифрових сервісів для всіх категорій користувачів, зокрема людей з інвалідністю, набуває особливої ваги. Розробка адаптивного інтерфейсу сприяє реалізації принципів інклюзивної освіти, цифрової рівності та соціальної інтеграції, що відповідає як національним, так і міжнародним стандартам.

Якість проведеного дослідження. У роботі продемонстровано системний підхід до вирішення поставленої задачі. Автором проведено аналіз сучасних технологій адаптивного дизайну, обґрунтовано вибір архітектури інформаційної системи, розроблено функціональні модулі з урахуванням потреб користувачів з різними типами порушень. Важливо, що дослідження охоплює як технічні, так і гуманітарні аспекти інклюзивного доступу до цифрових ресурсів. Робота має чітку структуру, логічну послідовність викладу та відповідає вимогам до кваліфікаційних робіт.

Позитивні риси кваліфікаційної бакалаврської роботи. Серед позитивних аспектів варто відзначити практичну спрямованість розробки, актуальність теми, грамотне використання сучасних інструментів програмування та веб-розробки. Автор продемонстрував вміння працювати з адаптивними технологіями, враховувати потреби користувачів з інвалідністю, а також здатність до самостійного аналізу та прийняття технічних рішень. Робота оформлена відповідно до встановлених стандартів, містить необхідні додатки та ілюстративний матеріал.

Недоліки. До незначних недоліків роботи можна віднести:

- в дослідженні представлений обмежений обсяг тестування системи на реальних користувачах з інклюзією, що зменшує емпіричну обґрунтованість висновків;
- відсутність структурованого порівняльного аналізу з аналогічними адаптивними системами, що зменшує аналітичну глибину дослідження;
- окремі стилістичні неточності та описки в тексті, які не впливають на загальну якість роботи, але потребують редакційного доопрацювання.

Практична значимість висновків і рекомендацій. Запропоновані в роботі рішення можуть бути використані для створення або вдосконалення інформаційних систем у закладах освіти, державних установах, громадських організаціях, що працюють з людьми з інклюзією. Розроблена система має потенціал для подальшого розвитку та масштабування, а її впровадження сприятиме підвищенню доступності цифрових сервісів і реалізації принципів інклюзивного середовища.

Рецензент: _____
(підпис)

Погореловська І.Д., к.е.н., доцент,
доцент кафедри комп'ютерних та інформаційних
технологій і систем факультету фінансів та
цифрових технологій Державного податкового
університету

« 3 » червня 2025 р.

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Інформаційна система з адаптивним інтерфейсом для
осіб з інклюзією»

Київ: кафедра інформаційних систем в економіці, 2025 р.

У роботі розглянуто процес проектування інформаційної системи з адаптивним інтерфейсом, орієнтованої на користувачів з інклюзією. Проаналізовано існуючі підходи до забезпечення доступності, сформовано функціональні та нефункціональні вимоги, побудовано діаграми прецедентів, класів, послідовності. Розроблено інтерактивний прототип мобільного застосунку в Figma, адаптований під користувачів з порушеннями зору, моторики та когнітивного сприйняття. Надано можливі варіанти архітектурної реалізації системи з використанням сучасних веб-технологій.

Кваліфікаційна бакалаврська робота складається з трьох розділів, логічно пов'язаних між собою.

У першому розділі було охарактеризовано предметну галузь — цифрові сервіси доставки — та виявлено її потенціал для впровадження адаптивних інтерфейсів в контексті інклюзивних систем. У другому розділі сформульовано бізнес-, функціональні та нефункціональні вимоги до інформаційної системи з профілем інклюзії. У межах третього розділу було реалізований ключовий компонент проекту — прототип мобільного застосунку DeliveryApp у середовищі Figma. Прототип охоплює повний цикл користувача: від реєстрації та вибору режиму доступності до перегляду товарів, управління кошиком, доставки й оплати. У висновках узагальнено підходи і проєктні рішення, орієнтовані на ІС для осіб з інклюзією.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 86 сторінок, 5 таблиць, 33 рисунки, список використаних джерел з 24 найменувань, 2 додатки.

«ІНФОРМАЦІЙНА СИСТЕМА З АДАПТИВНИМ ІНТЕРФЕЙСОМ ДЛЯ ОСІБ З ІНКЛЮЗІЄЮ»

Перелік ключових слів: інформаційна система, адаптивний інтерфейс, інклюзія, доступність, UX-дизайн, прототип, моделювання, користувач з інвалідністю.

Об'єктом дослідження є процеси розробки інформаційних систем, орієнтованих на користувачів з інклюзією — осіб із порушеннями зору, моторики та когнітивного сприйняття. Системи такого типу повинні забезпечувати рівний доступ до цифрових сервісів і враховувати особливості взаємодії користувачів з інтерфейсами.

Предметом дослідження є методи, інструменти та підходи до проєктування інтерфейсів, що адаптуються під індивідуальні потреби користувачів, зокрема реалізація візуальної, моторної та когнітивної доступності через прототипування й моделювання поведінки інтерфейсу.

Метою дослідження є розробка концепції інформаційної системи з адаптивним інтерфейсом для осіб з інклюзією, яка забезпечує комфортну взаємодію та відповідає сучасним стандартам доступності, з урахуванням рекомендацій WCAG та принципів UX-дизайну.

Завданнями кваліфікаційної бакалаврської роботи є:

- проаналізувати потреби користувачів з інвалідністю щодо цифрових сервісів;
- дослідити існуючі рішення в галузі доступності та інклюзії;
- сформулювати функціональні та нефункціональні вимоги до інформаційної системи;
- побудувати моделі предметної області та сценарії використання (Use Case, діаграми класів, послідовності, активності);
- розробити прототип мобільного додатку в Figma з адаптацією інтерфейсу;

- оцінити відповідність прототипу принципам доступності та провести UX-аналіз;

- надати бачення можливої реалізації інформаційної системи на основі сучасних вебтехнологій.

Апаратні засоби: при проектуванні та прототипуванні використовувався персональний комп'ютер Acer Nitro 5.

Використані методи дослідження: системний аналіз, моделювання (UML, SysML), візуальне прототипування, сценарний підхід до UX-дизайну, аналіз літературних джерел і нормативних стандартів.

Новизна роботи полягає в проектуванні концептуальної моделі адаптивного інтерфейсу, яка враховує особливості трьох груп користувачів: з порушенням зору, моторики та когнітивного сприйняття. Розроблений прототип може бути основою для подальшої реалізації інклюзивного цифрового продукту.

Стан реалізації: на даному етапі розроблено повноцінний інтерактивний прототип, підготовлено технічну документацію, моделі поведінки та структури системи. Програмна реалізація може бути здійснена з використанням стеку технологій React, Node.js, Firebase.

Практична значущість полягає у можливості використання розробленої моделі для створення адаптивних систем у сферах електронної комерції, медицини, освіти, що сприятиме цифровій інклюзії в Україні.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП	5
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ КОРИСТУВАЧІВ З ІНВАЛІДНІСТЮ.....	9
1.1 Характеристика цифрової інклюзії та об'єкта дослідження	9
1.2 Аналіз літературних джерел і практичного досвіду використання адаптивних інформаційних систем.....	11
РОЗДІЛ 2. РОЗРОБКА ВИМОГ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КОРИСТУВАЧІВ З ІНВАЛІДНІСТЮ	15
2.1 Аналіз і специфікація вимог до інформаційної системи	15
2.1 Бізнес-вимоги до системи	15
2.2 Функціональні вимоги до системи	17
2.1.3 Нефункціональні вимоги до системи	20
2.2 Постановка та алгоритм розв'язання задачі.....	21
2.2.1 Постановка задачі	21
2.2.2 Вихідні дані.....	23
2.2.3 Вхідні дані	25
2.2.4 Алгоритм розв'язання задачі.....	26
2.3 Моделювання інформаційної системи	27
2.3.2 Моделювання структури системи	30
РОЗДІЛ 3 ПРОЄКТУВАННЯ І РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	37
3.1 Інформаційне забезпечення	37
3.1.1 Загальна характеристика інформаційного забезпечення	37
3.1.2 Категорії даних, що відображаються в інтерфейсі.....	38
3.1.3 Способи отримання та оновлення даних в інтерфейсі (імітація в прототипі).....	41
ВИСНОВКИ	65
ДОДАТКИ	70

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – інтерфейс прикладного програмування.

CSS – каскадні таблиці стилів.

CRUD – створення, читання, оновлення, видалення (Create, Read, Update, Delete).

DB – база даних (Database).

Figma – графічний онлайн-редактор для UI/UX прототипування.

HTML – мова гіпертекстової розмітки.

ID – ідентифікатор.

JSON – JavaScript Object Notation.

JS – JavaScript.

SPA – односторінковий застосунок (Single Page Application).

SysML – системна мова моделювання.

UI – інтерфейс користувача.

UML – уніфікована мова моделювання.

UX – досвід користувача.

WCAG – Web Content Accessibility Guidelines (Веб-настанови з доступності контенту).

ВСТУП

Актуальність теми. У наш час, коли цифрові технології стають невід'ємною частиною повсякденного життя кожної людини, питання рівного доступу до цифрових ресурсів та інформаційних систем набуває дедалі більшої значущості. Розвиток інтернету, мобільних застосунків, електронного урядування, онлайн-освіти, цифрової медицини створює величезні можливості для комунікації, навчання та соціальної інтеграції. Проте ці можливості залишаються недоступними для значної частини населення — людей з інвалідністю, для яких неадаптовані інтерфейси й сервіси стають бар'єром, а не інструментом.

У світовій практиці вже накопичено значний досвід у сфері доступності цифрових продуктів, однак проблема залишається актуальною з кількох причин. По-перше, більшість рішень є універсальними, але не враховують індивідуальні особливості користувачів. По-друге, адаптивність інтерфейсів часто реалізована як додатковий функціонал, а не як базова складова архітектури системи. По-третє, темпи впровадження доступності в країнах, що розвиваються, зокрема в Україні, є повільними та нерівномірними. Саме тому розробка адаптивних інтерфейсів, які здатні автоматично реагувати на потреби конкретного користувача, є важливим і необхідним кроком.

Особливої актуальності дана тема набуває у контексті соціальної політики України, спрямованої на інтеграцію людей з інвалідністю в усі сфери життя. Забезпечення цифрової інклюзії є не лише вимогою часу, а й критерієм ефективності державної цифрової трансформації. Розробка інформаційної системи з адаптивним інтерфейсом, яка враховує потреби користувачів з різними формами інвалідності, є відповіддю на ці виклики. Такий підхід сприяє підвищенню рівня цифрової грамотності, самореалізації та соціалізації осіб з інклюзією, а також дозволяє реалізувати принципи універсального дизайну в повсякденній практиці розробки цифрових продуктів.

Аналіз останніх досліджень і публікацій. Протягом останніх десятиліть

проблема цифрової доступності стала об'єктом численних наукових досліджень і міжнародних проєктів. Значний внесок у формування теоретичних засад цієї галузі зробили провідні фахівці у сфері інформаційних технологій та людсько-комп'ютерної взаємодії. Зокрема, Тім Бернерс-Лі, один з батьків-засновників Інтернету, наголошував на важливості забезпечення рівного доступу до веб-контенту для всіх користувачів незалежно від їх фізичних або когнітивних можливостей. Його принцип "Web for All" став гаслом глобальної спільноти розробників, які підтримують ідеї відкритого та інклюзивного Інтернету.

Не менш важливими є праці Юзабіліті-експерта Якоба Нільсена, який акцентував увагу на принципах зручності, ефективності та доступності у проєктуванні інтерфейсів. Крім того, варто відзначити діяльність консорціуму W3C, який розробив керівні принципи WCAG (Web Content Accessibility Guidelines), що стали міжнародним стандартом для оцінювання рівня доступності вебсайтів.

Практична реалізація ідей доступності представлена у продуктах таких компаній, як Microsoft (Accessibility Suite), Apple (VoiceOver), Google (Lookout), що дозволяють компенсувати деякі форми інвалідності за допомогою програмних інструментів. Однак більшість з цих рішень не забезпечують комплексної адаптації інтерфейсу, яка б враховувала особливості користувача в режимі реального часу.

Вітчизняні дослідження переважно зосереджені на вузькоспеціалізованих аспектах: розробці екранних дикторів, збільшення шрифтів, голосовому керуванні, інтеграції допоміжних пристроїв. Разом з тим, спостерігається брак інтегрованих рішень, які поєднують різні засоби доступності в одну гнучку систему, що змінює поведінку в залежності від контексту. Тому ця кваліфікаційна робота спрямована на вирішення зазначених прогалин у науковій та практичній площині.

Метою дослідження є розробка концепції інформаційної системи з адаптивним інтерфейсом для осіб з інклюзією з урахуванням типу інвалідності, поведінкових моделей, яка забезпечує персоналізований доступ

до цифрових сервісів.

Основні завдання дослідження:

1. Провести огляд літератури та проаналізувати сучасний стан розробки інклюзивних цифрових систем.
2. Визначити функціональні та нефункціональні вимоги до адаптивного інтерфейсу.
3. Побудувати модель предметної області з використанням методів системного аналізу.
4. Розробити UML- і SysML-діаграми, що відображають архітектуру та динаміку роботи системи.
5. Створити прототип користувацького інтерфейсу з використанням інструментів UX-дизайну.
6. Реалізувати базову функціональність адаптивної логіки у вигляді інтерактивного прототипу.
7. Провести тестування прототипу на основі типових сценаріїв користувача.

Об'єкт дослідження. Процес взаємодії користувачів з інклюзією з інформаційними системами.

Предмет дослідження. Методи та інструменти розробки адаптивного інтерфейсу для користувачів з інвалідністю.

Методи дослідження. У рамках виконання даного дослідження були використані наступні методи:

- методи системного аналізу для моделювання предметної області;
- UML- і SysML-діаграми для структурного й поведінкового опису;
- методи UX-дизайну та прототипування (Figma);
- модульне, інтеграційне, сценарне та юзабіліті-тестування.

Теоретична, методична та практична значущість. Теоретична значущість дослідження полягає в поглибленні наукового розуміння принципів адаптивності в людсько-комп'ютерній взаємодії та уточненні класифікації засобів доступності.

Методична значущість виражається у формуванні системи критеріїв оцінювання інтерфейсів з точки зору інклюзивності, а також у розробці

підходів до моделювання користувацького досвіду осіб з обмеженими можливостями.

Практична значущість зосереджена на створенні реального прототипу, який може бути використаний як основа для розробки повнофункціонального застосунку, що забезпечує доступ до цифрових сервісів широкому колу користувачів з особливими потребами.

Інформаційна база дослідження. Під час виконання дослідження було використано низку джерел, що охоплюють як теоретичні, так і практичні аспекти розробки адаптивних інтерфейсів:

- публікації W3C щодо WCAG 2.1,
- технічна документація Microsoft, Google, Apple щодо реалізації функцій доступності,
- праці з юзабіліті та UX-дизайну зокрема роботи Якоба Нільсена та Дона Нормана,
- законодавчі та нормативні акти України щодо цифрової інклюзії,
- статистичні дані з офіційних сайтів Державної служби статистики України, звіти міжнародних організацій, які займаються правами осіб з інвалідністю.

Структура роботи. Загальний обсяг роботи становить 86 сторінок друкованого тексту, 33 рисунки на 21 сторінці, 5 таблиць на 9 сторінках, 2 додатки на 11 сторінках. Список використаних джерел налічує 24 найменування.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ КОРИСТУВАЧІВ З ІНВАЛІДНІСТЮ

1.1 Характеристика цифрової інклюзії та об'єкта дослідження

У сучасному суспільстві цифрові технології стали не лише зручним інструментом для виконання щоденних завдань, але й невід'ємною частиною соціального та професійного життя. Від доступу до освітніх платформ і медичних послуг до здійснення банківських операцій і участі в державному управлінні — майже кожен аспект життєдіяльності сьогодні так чи інакше пов'язаний із цифровими інформаційними системами. Однак для значної частини населення, зокрема осіб з інвалідністю, повноцінне використання цих систем залишається проблематичним.

Предметна галузь цього дослідження охоплює інформаційні технології, зосереджені на розробці адаптивних інтерфейсів користувача, з урахуванням специфічних потреб людей з інклюзією. Особлива увага приділяється питанням забезпечення доступності цифрових продуктів, дотриманню принципів універсального дизайну, а також створенню умов, за яких інтерфейс системи може змінювати свою структуру, елементи управління, розміри шрифтів, кольорову гаму тощо відповідно до індивідуальних потреб користувача.

Варто зазначити, що сучасні інформаційні системи, хоча й демонструють високий рівень автоматизації та функціональності, часто не враховують факторів інклюзії у своїй архітектурі. У багатьох випадках користувачі з інвалідністю змушені покладатися на сторонні рішення — допоміжні програми або спеціалізовані пристрої, які не завжди забезпечують повноцінну інтеграцію в цифрове середовище. Це створює бар'єри не лише у спілкуванні з технологіями, а й у соціальній адаптації таких осіб.

Об'єктом дослідження в межах даної роботи виступає процес взаємодії користувача з інформаційною системою з урахуванням його функціональних обмежень. Такий підхід дозволяє комплексно аналізувати фактори, які впливають на зручність, ефективність та доступність інтерфейсу для людей з різними типами інвалідності: порушеннями зору, моторики та когнітивних функцій.

Зважаючи на динамічний розвиток цифрової сфери, зростання кількості електронних сервісів і підвищення уваги до питань соціальної відповідальності, актуальність створення таких адаптивних систем лише зростає. Таким чином, дана предметна галузь поєднує в собі технологічні інновації, соціальні виклики та гуманістичний підхід до проектування програмного забезпечення.

Таблиця 1.1 – Аналіз методів оптимізації

№	Метод оптимізації	Суть методу	Переваги	Обмеження / недоліки
1	Адаптація за контекстом використання	Автоматичне виявлення типу пристрою, навколишнього середовища, часу доби	Покращення UX, зручність без ручного налаштування	Може вимагати доступ до сенсорів або конфіденційної інформації
2	Індивідуалізація за поведінковим профілем	Система аналізує дії користувача та підлаштовує інтерфейс	Високий рівень персоналізації, гнучкість	Потрібен час на навчання моделі, ризики помилкової інтерпретації дій
3	Масштабування та контраст	Динамічна зміна шрифтів, кольорів, елементів	Доступність для користувачів із порушенням зору	Не охоплює всі види інвалідності, потребує налаштування
4	Голосові інтерфейси	Керування інтерфейсом голосом	Зручність для людей із моторними порушеннями	Обмеження в шумних умовах, можливі труднощі з розпізнаванням мови
5	Модульна структура інтерфейсу	Можливість змінювати, вмикати або вимикати частини інтерфейсу	Гнучкість, легке налаштування	Потребує продуманої архітектури

Продовження таблиці 1.1

6	Альтернативні способи навігації	Жести, клавіатура, погляд, перемикачі	Зниження навантаження, зручність	Вимагає підтримки обладнання та сумісності
7	Інтеграція з існуючими інструментами доступності	Сумісність з VoiceOver, TalkBack, лупами	Швидке впровадження, знайомі засоби	Обмежена кастомізація, залежність від ОС
8	Інтерфейс з підказками та інструкціями	Вбудовані пояснення, допоміжні повідомлення	Полегшує використання інтерфейсу	Може перевантажувати UI, викликати дратування

Джерело: сформовано автором на основі виконаного дослідження

Крім того, аналіз предметної галузі передбачає врахування міжнародного досвіду, стандартів цифрової доступності (зокрема, WCAG), а також державної політики України щодо забезпечення прав осіб з інвалідністю. Зважаючи на вимоги цифрової трансформації та впровадження принципів інклюзивності у сфері ІТ, необхідність створення універсальних інформаційних систем із гнучкими інтерфейсами є очевидною та надзвичайно важливою.

1.2 Аналіз літературних джерел і практичного досвіду використання адаптивних інформаційних систем

Розробка адаптивних інтерфейсів для осіб з інвалідністю ґрунтується як на теоретичних засадах дизайну взаємодії, так і на практичних реалізаціях у сучасних інформаційних системах. У науковій літературі дана тематика знаходиться на перетині кількох галузей знань: UX-дизайну, когнітивної психології, інформаційних технологій, соціальних наук та інклюзивної педагогіки.

Вагомий внесок у розвиток теоретичних основ користувацького досвіду (UX) зробили Якоб Нільсен (Jakob Nielsen) і Дон Норман (Don Norman). У своїх працях, зокрема в книгах "The Design of Everyday Things" та "Emotional

Design", Дон Норман акцентує увагу на важливості емоційного зв'язку між користувачем та інтерфейсом, а Нільсен у працях "Usability Engineering" і "Designing Web Usability" пропонує принципи ефективної взаємодії, які безпосередньо впливають на зручність та інтуїтивність використання інтерфейсу — критично важливі аспекти для людей з обмеженими можливостями.

У практичній площині, великі ІТ-компанії активно впроваджують технології доступності. Наприклад:

– Microsoft розвиває пакет функцій Microsoft Accessibility Suite, що охоплює екранний диктор, лупу, візуальні контрасти та підтримку сторонніх пристроїв.

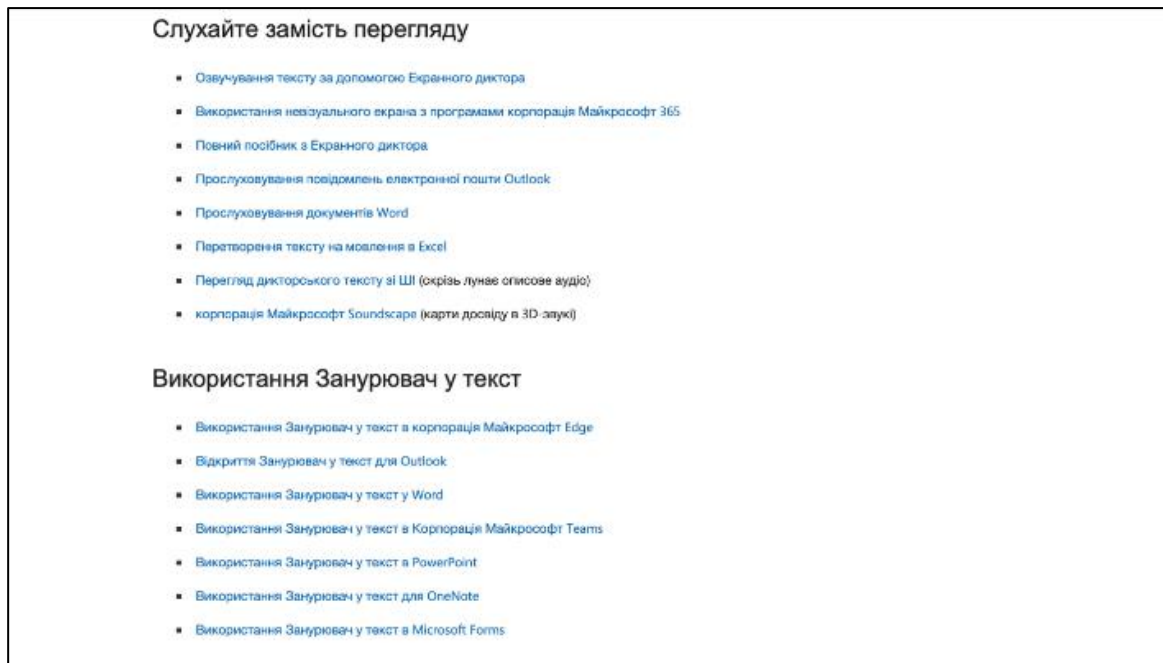


Рисунок 1.1 – Приклад пропозицій адаптації від компанії Microsoft

Джерело: знімок екрану створено автором

– Apple активно просуває свої рішення, такі як VoiceOver (екранний диктор для macOS/iOS), Switch Control, AssistiveTouch.

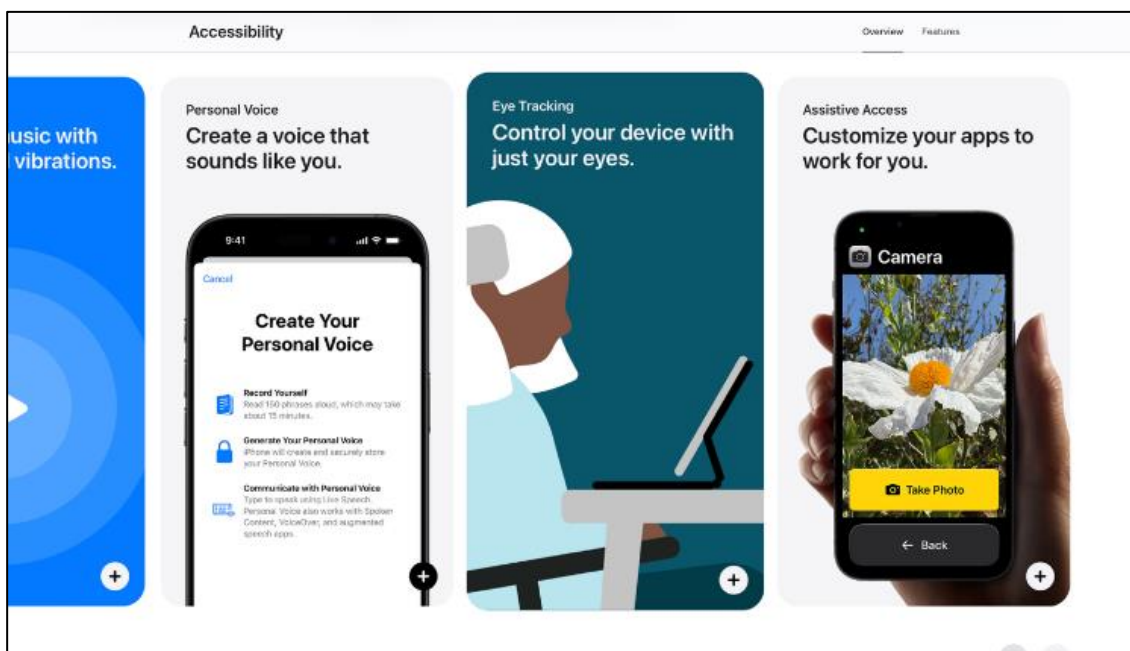


Рисунок 1.2 – Приклад пропозицій адаптації від компанії Apple (створення запису власного голосу для управління командами, контроль девайсів через керування очима, кастомізовані кнопки в додатках для зручного користування тощо.)

Джерело: знімок екрану створено автором

- Google, у свою чергу, розробив сервіси на кшталт Lookout та TalkBack для Android, які допомагають користувачам з порушеннями зору.

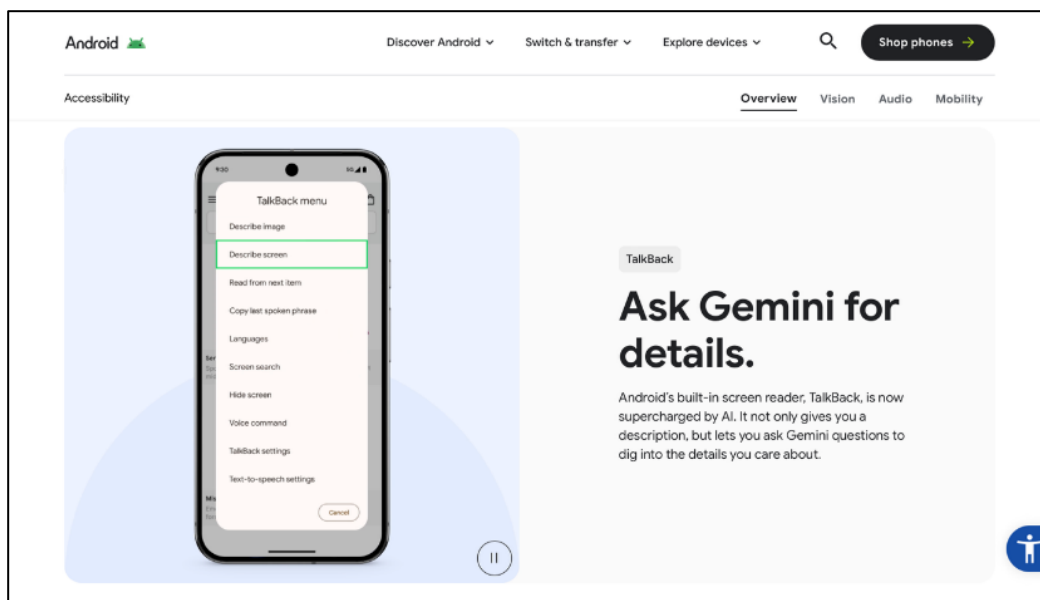


Рисунок 1.3 – Приклад пропозицій адаптації від компанії Google для Android з використанням Gemini

Джерело: знімок екрану створено автором

На практиці ці рішення дозволяють користувачам з інвалідністю повноцінно працювати зі смартфонами, ПК, сайтами та мобільними

застосунками. Водночас, незважаючи на значний прогрес, ці інтерфейси, як правило, потребують ручного налаштування та не адаптуються автоматично до конкретних потреб користувача. Саме тому розробка інтелектуальних адаптивних інтерфейсів, що самостійно підлаштовуються до виявленого типу інклюзії, залишається важливим і перспективним напрямом.

У вітчизняному контексті інтерес до цифрової інклюзії виявляється здебільшого в межах окремих наукових статей і дипломних робіт у галузі прикладної інформатики. Проте бракує масштабних проєктів або комплексних рішень, які б одночасно враховували потреби різних типів інвалідності. Також недостатньо досліджена інтеграція подібних рішень у вже наявні цифрові платформи державного або комерційного рівня.

Отже, аналіз літературних джерел і практичного досвіду свідчить про наявність теоретичної бази та технологічних рішень, які можуть бути основою для створення власної інформаційної системи з адаптивним інтерфейсом. Водночас, інноваційним елементом запропонованої розробки є комплексний підхід до автоматичної персоналізації, що дає змогу гнучко реагувати на індивідуальні потреби користувача без його активної участі в налаштуваннях.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ КОРИСТУВАЧІВ З ІНВАЛІДНІСТЮ

2.1 Аналіз і специфікація вимог до інформаційної системи

2.1 Бізнес-вимоги до системи

Усі вимоги до застосунку поділено на 3 групи: бізнес-вимоги, функціональні вимоги та нефункціональні вимоги.

Успішна розробка інформаційної системи з адаптивним інтерфейсом для осіб з інклюзією вимагає формального опису вимог, які визначають функціональність, продуктивність, інтерфейсні характеристики, надійність та інші параметри системи. Вимоги систематизовано за допомогою діаграм пакетів, що дозволяє чітко відобразити структуру, склад і взаємозв'язки всіх елементів моделі.

Бізнес-вимоги відображають стратегічні цілі, які має реалізувати система, а також загальні потреби зацікавлених сторін. Основною бізнес-ціллю є створення доступної, зручною у використанні інформаційної системи для осіб з інклюзією, яка автоматично адаптується до індивідуальних потреб користувачів з різними видами обмежень (порушення зору, слуху, опорно-рухового апарату, когнітивні особливості).

Ієрархія бізнес-вимог побудована на базі діаграми вимог SysML типу *requirement diagram*, де показано зв'язки між джерелами вимог (користувачі, нормативно-правові акти, технічні стандарти), бізнес-правилами (та обмеженнями (наявність підтримки лише певних типів пристроїв на етапі MVP).

Ключові бізнес-вимоги включають:

- Забезпечення універсального доступу до цифрових послуг для осіб з інклюзією.

- Підвищення ефективності цифрової взаємодії за рахунок адаптації інтерфейсу.
- Створення гнучкої архітектури системи, здатної до масштабування.
- Відповідність чинним міжнародним стандартам з цифрової доступності.
- Підтримка декількох мов інтерфейсу з урахуванням мовних уподобань користувачів.

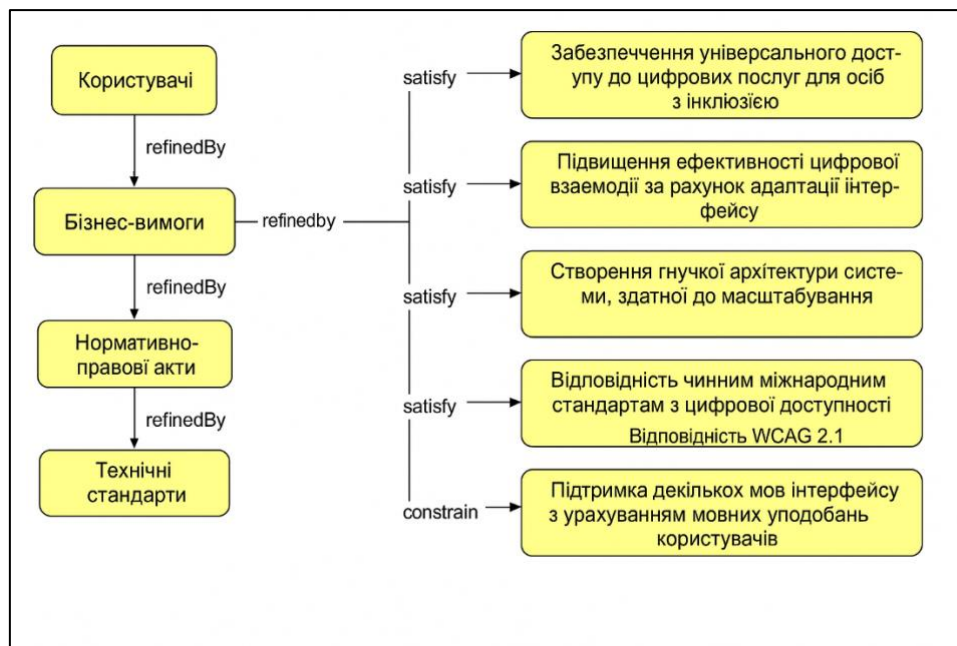


Рисунок 2.1. – Діаграма "Ієрархія бізнес-вимог "

Джерело: розроблено автором на основі виконаного дослідження

У діаграмі пакетів вимоги згруповано за тематичними блоками: «Доступність», «Персоналізація», «Інтеграція з іншими сервісами», «Законодавче регулювання», «Мультимовність».

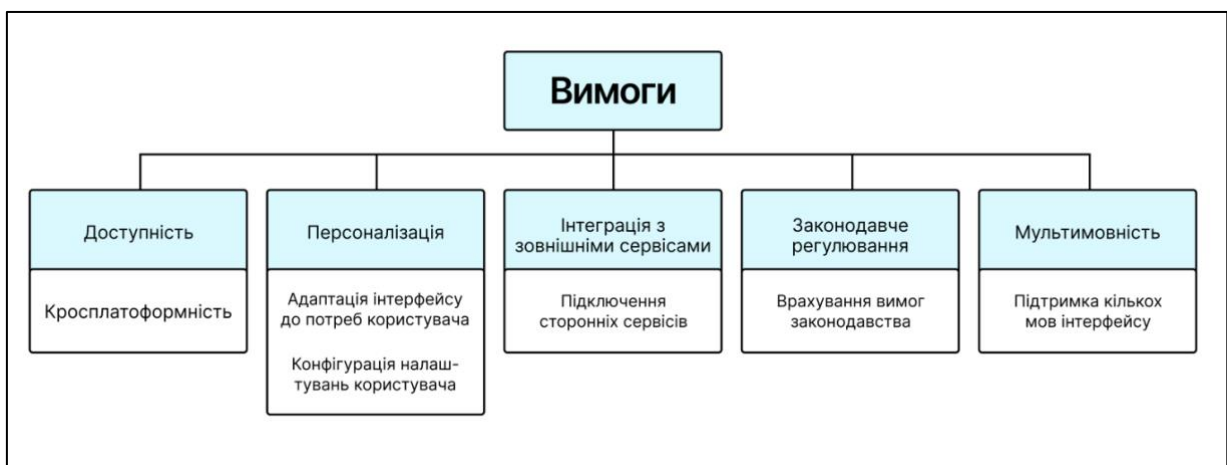


Рисунок 2.2. – Діаграма пакетів вимоги

Джерело: розроблено автором на основі виконаного дослідження

Для формалізації вимог було використано таблицю специфікацій бізнес-вимог.

Таблиця 2.1 – Специфікація бізнес-вимог

Ідентифікатор	Назва	Повна назва	Тип	Статус	Складність	Пріоритет	Ризик	Примітка
BR-01	Універсальний доступ	Система має забезпечити доступність для користувачів із різними порушеннями	Бізнес	Затверджено	Висока	Високий	Середній	Визначається за WCAG 2.1
BR-02	Мультимовність	Інтерфейс повинен підтримувати щонайменше 3 мови (українську, англійську, польську)	Бізнес	Опрацьовується	Середня	Середній	Низький	Дозволить масштабувати систему на інші регіони
BR-03	Юридична відповідність	Система має відповідати нормативно-правовим актам України щодо цифрової інклюзії						

Джерело: розроблено автором на основі виконаного дослідження

2.2 Функціональні вимоги до системи

Функціональні вимоги визначають, що саме має робити система, якими мають бути її реакції на вхідні дії та як вона має взаємодіяти з користувачем і зовнішніми системами. У контексті інклюзивної системи, особлива увага приділяється не лише бізнес-логіці (вибір супермаркету, замовлення товару), але й адаптивності інтерфейсу: масштаб, озвучення, навігація тощо.

У цьому проєкті визначено 8 логічних блоків із відповідними підвимогами (усього — 22 функціональні вимоги):

1. Реєстрація та вхід:

- Початковий екран застосунку: Система повинна відобразити головний екран з логотипом, слоганом, вітальним текстом, та кнопкою “Start” при першому запуску застосунку;
- Перехід до реєстрації: Після натискання на кнопку “Start” система повинна відкрити форму реєстрації з полями для імені, e-mail та паролю;

- Перехід до налаштувань доступності: Після успішної реєстрації система повинна автоматично перевести користувача до екрана налаштувань доступності.

2. Доступність:

- Вибір адаптивного режиму: Користувач повинен мати можливість обрати один з режимів доступності: порушення зору, моторики або когнітивні особливості;
- Активація адаптивного інтерфейсу: Після вибору режиму система повинна активувати відповідні UI-зміни: високий контраст, озвучення, збільшені елементи, спрощена навігація тощо;

3. Вибір супермаркету:

- Екран вибору магазину: Система повинна відобразити логотипи супермаркетів (Сільпо, Novus, Metro тощо) для вибору після налаштування доступності;
- Перехід до каталогу магазину: При натисканні на логотип магазину система повинна показати каталог товарів обраного супермаркету.

4. Перегляд товарів:

- Каталог товарів: Система повинна відображати товари у вигляді плиток з фото, назвою, ціною та кнопкою “додати”;
- Фільтр товарів: У верхній частині каталогу має бути фільтр для вибору категорій товарів;
- Додавання товару до кошика: Після натискання кнопки “+” на плитці товару, товар повинен додаватися до кошика, а індикатор оновлювати кількість.

5. Кошик:

- Доступ до кошика: Іконка кошика повинна бути завжди доступною в нижньому правому куті навігаційної панелі;
- Вміст кошика: При відкритті кошика система повинна показувати перелік товарів з назвою, кількістю, ціною кожного та загальною сумою;

- Редагування кошика: Користувач повинен мати можливість змінити кількість товару або видалити його з кошика кнопками “+”, “-” чи “видалити”.

6. Оформлення замовлення:

- Перехід до доставки: Після натискання “Complete order” користувач переходить до екрана оформлення доставки;
- Дані доставки: Користувач вводить адресу, обирає тип доставки (кур’єр / самовивіз) та час доставки;
- Метод оплати: Перед підтвердженням замовлення користувач обирає метод оплати (готівка або картка).

7. Інші вимоги:

- Адаптація інтерфейсу: Всі екрани повинні змінюватися відповідно до вибраного режиму доступності;
- Озвучення дій: Якщо активоване озвучення, система повинна озвучувати ключові дії та елементи інтерфейсу.

Діаграма вимог SysML для функціонального блоку містить елементи типу «requirement», зв’язані з функціями системи, зокрема адаптивними модулями, модулями обробки даних, інтерфейсними засобами, адміністративними компонентами.

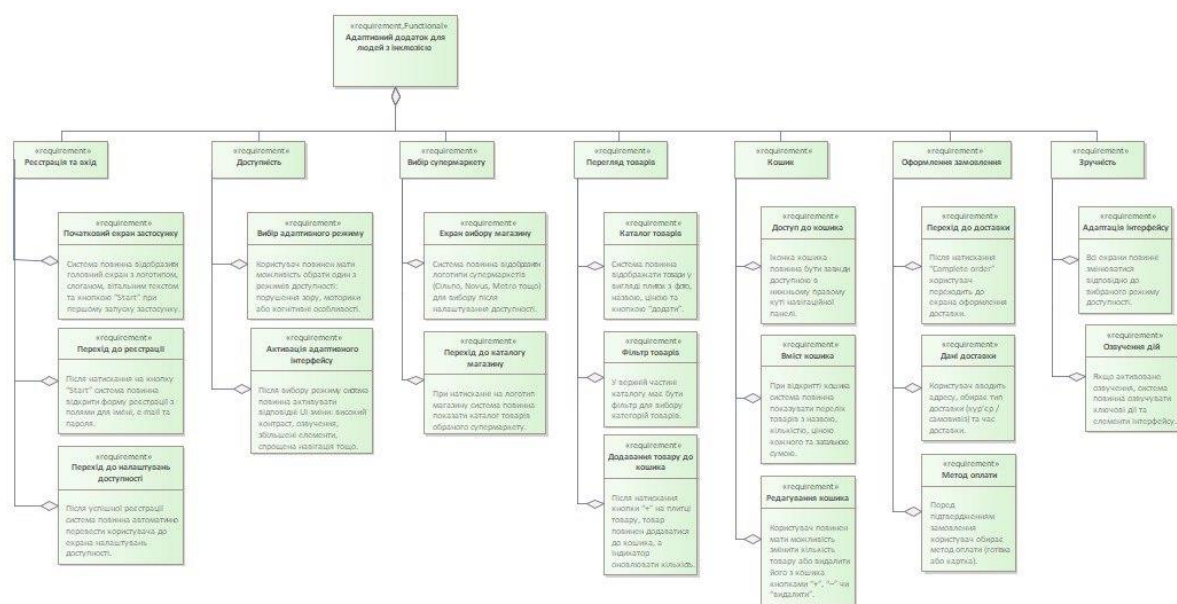


Рисунок 2.3. – Діаграма "Функціональні вимоги"

Джерело: розроблено автором на основі виконаного дослідження

Для систематизації створено окрему діаграму пакетів, де функціональні вимоги поділено на: «Реєстрація та автентифікація», «Адаптація інтерфейсу», «Комунікація з користувачем», «Обробка даних», «Інтеграція».

2.1.3 Нефункціональні вимоги до системи

Нефункціональні вимоги описують якісні характеристики системи, зокрема її продуктивність, надійність, безпеку, масштабованість, зручність використання, відповідність стандартам. Основні вимоги цієї групи:

- Висока доступність системи (не менше 99% часу в мережі).
- Можливість масштабування інтерфейсу під різні пристрої (адаптивна верстка).
- Захист персональних даних згідно з GDPR і національним законодавством.
- Час відгуку інтерфейсу – не більше 1 секунди при стандартному навантаженні.
- Надійність і безперебійність при роботі з допоміжними технологіями.
- Простота використання, оцінена за метрикою SUS (System Usability Scale) – не нижче 85.

Діаграма вимог SysML відображає взаємозв'язки між нефункціональними вимогами та відповідними цілями, припущеннями, обмеженнями.

До нефункціональних вимог належать:

1. Одноручна зручність: Інтерфейс повинен бути оптимізований для використання однією рукою на екранах 4.7–6.8 дюймів.
2. Швидкодія інтерфейсу: Час завантаження будь-якого екрана не повинен перевищувати 2 секунд при стабільному інтернеті.
3. Висока доступність системи: Додаток має забезпечувати не менше 99.5% доступності протягом місяця.
4. Сумісність з ОС: Підтримка останніх 3 версій iOS та Android, з повідомленням про обмеження для застарілих версій.

5. Захист даних: Користувацькі дані повинні зберігатись у зашифрованому вигляді (AES-256).

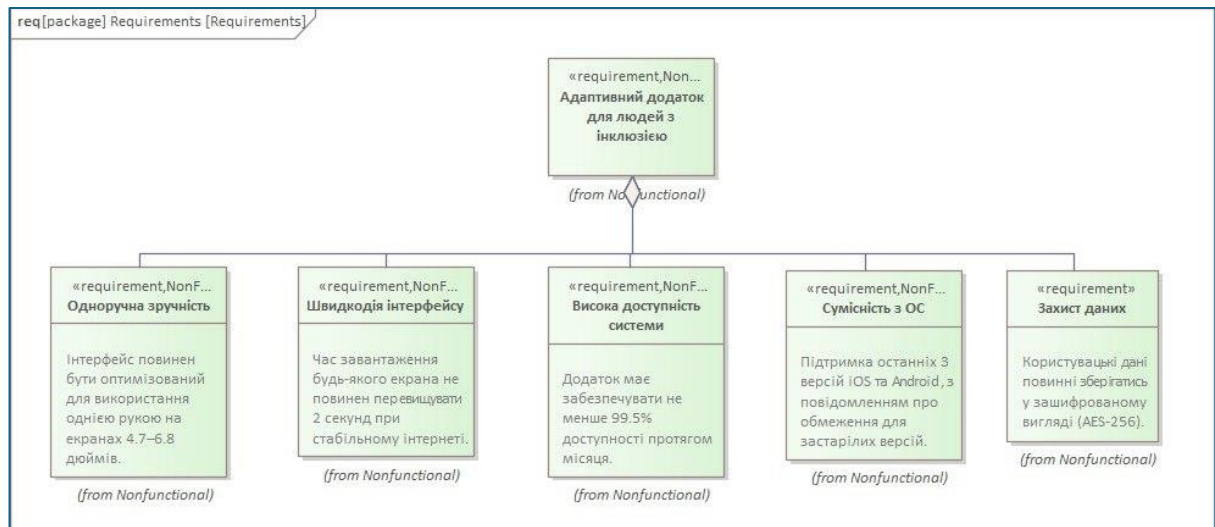


Рисунок 2.4 – Діаграма "Нефункціональні вимоги"

Джерело: розроблено автором на основі виконаного дослідження

Завдяки систематизації вимог і побудові відповідних діаграм стало можливим ефективно спланувати подальший хід розробки, розподілити обов'язки в команді, оцінити ризики та спрогнозувати часові рамки впровадження.

Сформульовані вимоги відповідають усім основним критеріям якості: вони однозначні, перевірні, чіткі, прості, зрозумілі для всіх учасників розробки. Вимоги взаємно несуперечливі, повні, необхідні, атомарні та незалежні. Така сукупність вимог гарантує стабільність архітектури системи, ефективність процесів її реалізації та відповідність очікуванням цільових користувачів.

2.2 Постановка та алгоритм розв'язання задачі

2.2.1 Постановка задачі

У межах дослідження ставиться задача розробки інформаційної системи, здатної забезпечити персоналізовану адаптацію інтерфейсу для користувачів з

інклюзією. Проблематика цифрової нерівності вимагає створення рішень, які не лише враховують технічні обмеження користувачів, але й здатні автоматично або вручну перебудувувати інтерфейс під конкретні потреби.

Система повинна обслуговувати користувачів з порушеннями зору, моторики, когнітивного сприйняття, забезпечуючи:

- автоматичне визначення типу інклюзії або ручний вибір режиму доступності;
- застосування шаблонів адаптивного відображення UI-елементів;
- підтримку альтернативного введення/виводу (озвучення, голосові команди, великі кнопки тощо);
- інтуїтивно зрозумілу структуру мобільного застосунку, з урахуванням обмежень у навігації та сприйнятті.

У якості тестового сценарію було обрано мобільний додаток для доставки товарів із супермаркетів — DeliveryApp. Його функціональність охоплює: реєстрацію, вибір режиму доступності, перегляд товарів, управління кошиком, оформлення замовлення. Для кожного кроку користувацького шляху мають бути забезпечені UI-механізми, що адаптуються до конкретного режиму доступності.

Таким чином, мета задачі — побудувати систему, що забезпечить:

- можливість початкової і подальшої адаптації інтерфейсу;
- інтуїтивність і простоту взаємодії для всіх категорій користувачів;
- відповідність сучасним стандартам доступності (зокрема, WCAG).

Розв'язання задачі охоплює кілька рівнів:

1. Концептуальний рівень — постановка вимог і сценаріїв використання.
2. Моделювання логіки — створення діаграм прецедентів, послідовності, активності, класів.
3. Прототипування — створення UI-макетів у Figma з варіантами адаптації.
4. Валідація — проведення юзабіліті-аналізу та верифікація відповідності вимогам.

Усі ці кроки спрямовані на вирішення основної задачі — забезпечити ефективну та зручну взаємодію користувача з адаптивною системою,

враховуючи його особливості.

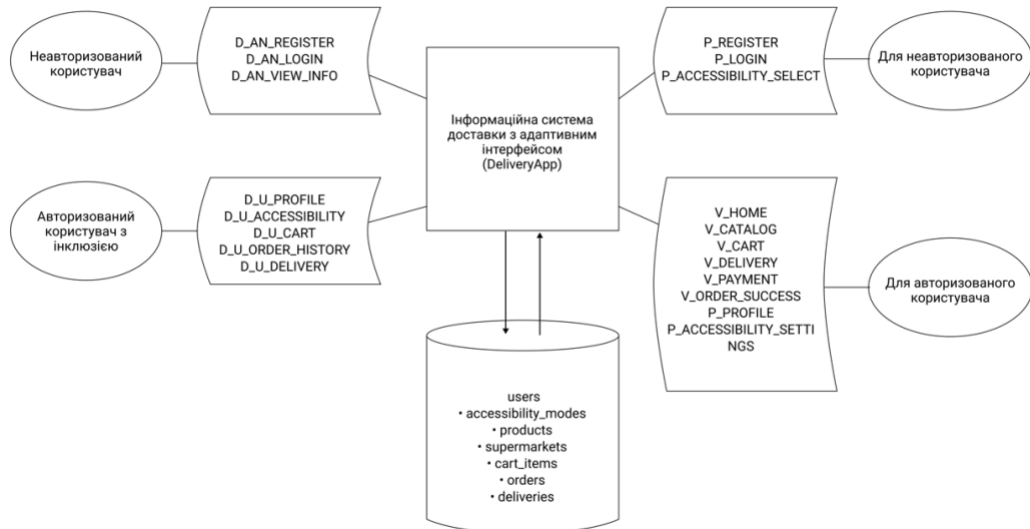


Рисунок 2.5 Інформаційна модель задачі для системи DeliveryApp

Джерело: розроблено автором на основі виконаного дослідження

2.2.2 Вихідні дані

Вихідна інформація системи складається з результатів взаємодії користувача з адаптивним інтерфейсом мобільного застосунку. Вона охоплює як підтвердження виконаних дій (вхід, реєстрація, оформлення замовлення), так і візуальні представлення даних (каталог товарів, інформація про доставку, стан кошика), що оновлюються відповідно до обраного режиму доступності.

До вихідної інформації належать:

- повідомлення про успішну авторизацію чи реєстрацію користувача;
- відображення головного екрану з урахуванням адаптаційних налаштувань;
- результати пошуку товарів;
- виведення вмісту кошика й деталей доставки;
- повідомлення про помилки або успішне виконання замовлення;
- інформаційні екрани для перегляду профілю, історії замовлень тощо.

Особливістю формування вихідних даних у DeliveryApp є адаптивна логіка інтерфейсу. Наприклад, одна й та сама інформація (перелік товарів або форма оформлення замовлення) може бути представлена в різному форматі — зі збільшеним шрифтом, скороченим текстом або озвучуванням — відповідно

до обраного режиму доступності: візуального, моторного або когнітивного.

Детальніше основні типи вихідних повідомлень та форм їх подання наведено в таблиці 2.2.

Таблиця 2.2 – Специфікація бізнес-вимог

Назва вихідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надання	Призначення
Підтвердження входу в систему	O_LOGIN_SUCCESS	Повідомлення	Миттєво після входу	Неавторизований користувач
Підтвердження реєстрації	O_REGISTER_DONE	Повідомлення	Миттєво після реєстрації	Неавторизований користувач
Підтвердження вибору режиму доступності	O_ACCESS_SET	Повідомлення	Одразу після вибору режиму	Авторизований користувач
Відображення головної сторінки	O_HOME_VIEW	Екранна форма	Після входу в систему	Авторизований користувач
Каталог товарів	O_CATALOG_DATA	Екранна форма	При переході на каталог	Авторизований користувач
Вміст кошика	O_CART_DATA	Екранна форма	У режимі замовлення	Авторизований користувач
Підтвердження замовлення	O_ORDER_SUCCESS	Повідомлення	Миттєво після оформлення	Авторизований користувач
Інформація про доставку	O_DELIVERY_INFO	Екранна форма	При виборі способу доставки	Авторизований користувач
Історія замовлень	O_ORDER_HISTORY	Екранна форма	За запитом	Авторизований користувач
Налаштування профілю	O_PROFILE_VIEW	Екранна форма	При переході до профілю	Авторизований користувач

Джерело: розроблено автором на основі виконаного дослідження

2.2.3 Вхідні дані

Вхідна інформація є основою для функціонування адаптивної інформаційної системи та забезпечує її реакцію на дії користувача. У контексті системи доставки з адаптивним інтерфейсом (DeliveryApp) до вхідних даних належать як безпосередні дії користувачів, так і введені значення в формах, запити на зміну інтерфейсу чи перегляд певної інформації.

Найбільш типовими вхідними даними є: заповнення реєстраційної форми, авторизація, вибір режиму доступності, додавання товарів до кошика, вибір способу доставки та оплати, підтвердження замовлення тощо. Ці дії ініціюють відповідні сценарії взаємодії, на які система має реагувати в режимі реального часу. Кожна група користувачів (неавторизований, авторизований, користувач з інклюзією) вносить свій набір вхідних даних, які відображають їх потреби та логіку використання системи. Наприклад, користувач з інклюзією передає системі додаткову інформацію про тип обраного режиму доступності, що впливає на подальше відображення інтерфейсу.

Докладніше типи вхідних даних представлено в таблиці 2.3.

Таблиця 2.2 – Специфікація бізнес-вимог

<i>Назва вхідних даних</i>	<i>Ідентифікатор</i>	<i>Форма подання</i>	<i>Джерело</i>	<i>Призначення</i>
Дані реєстрації користувача	I_REGISTER	Форма	Неавторизований користувач	Створення облікового запису
Облікові дані (логін, пароль)	I_LOGIN	Форма	Неавторизований користувач	Авторизація користувача
Вибір режиму доступності	I_ACCESS_MODE	Кнопка / перемикач	Усі користувачі	Авторизація користувача
Додавання товару до кошика	I_ADD_TO_CART	Кнопка	Авторизований користувач	Формування замовлення
Вибір товару для перегляду	I_VIEW_PRODUCT	Натискання	Авторизований користувач	Перехід до опису товару

Продовження таблиці 2.2

Вибір методу доставки	I_DELIVERY_SELECT	Радіо-кнопка / список	Авторизований користувач	Формування способу отримання
Вибір способу оплати	I_PAYMENT_SELECT	Радіо-кнопка / список	Авторизований користувач	Вибір способу оплати
Підтвердження замовлення	I_CONFIRM_ORDER	Кнопка	Авторизований користувач	Завершення покупки
Зміна налаштувань профілю	I_PROFILE_UPDATE	Форма	Авторизований користувач	Редагування персональних даних
Вибір мови / теми (доступності)	I_UI_SETTINGS	Перемикачі	Авторизований користувач	Налаштування інтерфейсу

Джерело: розроблено автором на основі виконаного дослідження

2.2.4 Алгоритм розв'язання задачі

Для ефективної реалізації інформаційної системи з адаптивним інтерфейсом було сформовано алгоритм, який охоплює основні етапи взаємодії користувача з системою, починаючи з першого входу до моменту оформлення замовлення. Основна ідея алгоритму полягає в тому, що система повинна автоматично або вручну адаптувати інтерфейс під потреби користувача з інклюзією на кожному з етапів взаємодії.

Алгоритм розв'язання задачі містить наступні кроки:

- КРОК 1. Запуск застосунку. Користувач відкриває застосунок, система показує вітальний екран з кнопкою «Start».
- КРОК 2. Реєстрація або авторизація. Після натискання кнопки користувач заповнює форму реєстрації (ім'я, e-mail, пароль) або входить у систему.
- КРОК 3. Налаштування доступності. Система пропонує обрати адаптивний режим:
 - для людей з порушенням зору (високий контраст, озвучення);
 - для користувачів із порушеннями моторики (великі кнопки);
 - для когнітивних особливостей (спрощена навігація).
- КРОК 4. Застосування обраного режиму. Інтерфейс змінюється згідно з

вибраним режимом: змінюються шрифти, кольори, структура елементів тощо.

- КРОК 5. Вибір супермаркету. Користувач обирає магазин із запропонованого списку (логотипи, назви).
- КРОК 6. Перегляд товарів. Система показує каталог з фільтрами, кнопками додавання та візуальними/аудіо підказками.
- КРОК 7. Робота з кошиком. Користувач може додавати, змінювати кількість або видаляти товари, переглядати загальну суму.
- КРОК 8. Оформлення замовлення. Користувач переходить до оформлення: вводить адресу, обирає тип доставки, спосіб оплати.
- КРОК 9. Підтвердження та завершення замовлення. Система озвучує або виводить повідомлення про успішне оформлення замовлення.

2.3 Моделювання інформаційної системи

2.3.1 Моделювання поведінки системи

З метою побудови моделі інформаційної підсистеми було прийнято рішення використовувати об'єктно-орієнтований підхід до моделювання. Цей метод дозволяє ефективно працювати з комплексними системами, полегшуючи їхнє розуміння за рахунок узагальнення та абстрагування від зайвих деталей. Об'єктний підхід надає змогу зосередитися на суттєвих аспектах функціонування системи, не відволікаючись на технічні нюанси, що не мають ключового значення на етапі концептуального проектування. Для реалізації моделювання було використано програмне середовище Enterprise Architect, яке є одним з найпоширеніших і найзручніших інструментів для побудови UML-діаграм різного рівня складності.

У процесі моделювання першочергово було створено діаграму прецедентів (варіантів використання) — англійською Use Case Diagram. Вона є важливим етапом у розробці системи, оскільки дозволяє візуалізувати всі

можливі сценарії взаємодії користувача з системою. У контексті UML така діаграма відображає основні функції, які система повинна виконувати, та користувачів, які ініціюють ці дії. Дана діаграма наочно демонструє зв'язки між акторами — тобто суб'єктами, що взаємодіють із системою, — та відповідними прецедентами — окремими функціональними можливостями системи. На основі отриманої інформації була побудована відповідна модель, що дозволяє зосередитися на основних можливостях системи, орієнтованих на кінцевого користувача (рис. А.1).

Наступним етапом стало створення діаграм послідовності (англ. Sequence Diagrams), які є одним із ключових елементів опису динамічної поведінки системи. Цей тип UML-діаграми використовується для відображення логіки взаємодії між окремими об'єктами протягом певного проміжку часу. Кожна діаграма ілюструє, як саме відбувається обмін повідомленнями, які об'єкти ініціюють ті чи інші дії та у якій послідовності. Це дозволяє краще зрозуміти механізм реалізації функціональності, описаної у відповідному прецеденті.

Спираючись на раніше побудовану діаграму прецедентів (рис. А.1), було здійснено деталізацію кожного варіанта використання через створення відповідних діаграм послідовності. Це дозволяє чітко побачити, як саме реалізуються функції системи, яким чином відбувається взаємодія між користувачем і внутрішніми компонентами системи, і яким є порядок виконання дій. У додатку А наведено приклади основних сценаріїв взаємодії користувачів із системою, кожен з яких відображений у вигляді окремої діаграми послідовності, що сприяє глибшому розумінню логіки роботи системи загалом. Розглянемо ключові сценарії взаємодії користувачів із системою, представлені на діаграмах послідовності:

- Рис. А.2 «Запуск застосунку та виведення привітального екрана» демонструє початкову взаємодію користувача із системою. При відкритті застосунку користувач ініціює виклик `open app()`, після чого інтерфейс користувача (App UI) реагує, викликаючи метод `displayWelcomeScreen()` — виводить привітальний екран із логотипом і кнопкою початку.

- *Рис. А.3* «Реєстрація нового користувача та перехід до налаштувань доступності» ілюструє процес взаємодії між користувачем, інтерфейсом (App UI), сервісом авторизації (AuthService), базою даних (Database) і екраном налаштувань доступності. Після натискання кнопки Start, користувач надсилає дані до App UI, який ініціює запит `sendRegistrationData()` до AuthService. Останній створює новий обліковий запис у базі даних через `createUserAccount()` та отримує відповідь `returnSuccess()`. Після підтвердження успішної реєстрації, інтерфейс виконує `goToAccessibilitySettings()` — перехід до екрану налаштувань доступності.
- *Рис.А.4* «Активація адаптивного режиму інтерфейсу» відображає послідовність дій під час вибору користувачем режиму доступності. Після вибору параметра (наприклад, для людей з порушенням зору), екран налаштувань (AccessibilityScreen) ініціює метод `activate adaptive interface()` до App UI. Інтерфейс відповідно змінює вигляд системи за допомогою `update interface appearance()`, щоб відобразити адаптовану версію згідно з обраними параметрами користувача.
- *Рис.А.5* "Процес відкриття каталогу супермаркету" демонструє послідовність взаємодій між користувачем та системою для доступу до каталогу обраного супермаркету. Діаграма відображає логіку обробки запитів, починаючи з перегляду користувачем списку супермаркетів, його подальшого вибору шляхом натискання на логотип супермаркету, і завершуючи відкриттям відповідного каталогу в інтерфейсі користувача додатка.
- *Рис.А.6* "Процес відображення списку товарів" демонструє послідовність взаємодій між екраном каталогу (CatalogScreen) та сервісом продуктів (ProductService) для отримання та відображення списку товарів. Діаграма відображає логіку обробки запитів, починаючи з запиту екраном каталогу списку товарів у сервісу продуктів, отримання відповіді з продуктами від сервісу, і завершуючи відображенням цих товарів користувачеві у вигляді плиток.

- *Рис.А.7* "Процес фільтрації товарів за категоріями" демонструє послідовність взаємодій між користувачем, екраном каталогу (CatalogScreen) та компонентом фільтрації (FilterComponent) для відображення відфільтрованих результатів. Діаграма відображає логіку обробки запитів, починаючи з вибору користувачем категорії, подальшого застосування фільтра компонентом фільтрації, і завершуючи відображенням екраном каталогу відфільтрованих результатів користувачеві.
- *Рис.А.8* "Процес додавання товару до кошика" демонструє послідовність взаємодій між користувачем, екраном каталогу (CatalogScreen) та сервісом кошика (CartService) для додавання товару до кошика. Діаграма відображає логіку обробки запитів, починаючи з натискання користувачем кнопки "+", подальшого додавання продукту до кошика через сервіс кошика, підтвердження додавання, і завершуючи оновленням відображення кількості товару на екрані для користувача.
- *Рис.А.9* "Процес відображення вмісту кошика" демонструє послідовність взаємодій між користувачем, інтерфейсом додатку (App UI) та сервісом кошика (CartService) для відображення вмісту кошика. Діаграма відображає логіку обробки запитів, починаючи з натискання користувачем на іконку кошика, подальшого отримання вмісту кошика від сервісу кошика, повернення списку товарів, і завершуючи відображенням кошика користувачеві в інтерфейсі додатка.

2.3.2 Моделювання структури системи

Розробка інформаційної системи з адаптивним інтерфейсом для осіб з інвалідністю передбачає формування чіткої, логічно впорядкованої структури, яка дозволить забезпечити як ефективну обробку даних, так і зручну взаємодію з кінцевими користувачами. Для моделювання структури системи застосовується система структурних діаграм на основі специфікації SysML та UML. У цьому підрозділі буде представлено ієрархію блоків, зв'язки між

внутрішніми компонентами, потоками та класами, а також структурні класи системи відповідно до прийнятих стереотипів.

Діаграма визначення блоків: "Ієрархічна структура системи" наведена у Додатку А на малюнку А.10. На верхньому рівні структура системи складається з трьох основних блоків:

- Технічне забезпечення (Hardware);
- Програмне забезпечення (Software);
- Середовище користувача (User Environment).

Блок *Технічне забезпечення* включає серверну інфраструктуру (сервер бази даних, сервер додатку), клієнтські пристрої (ПК, планшети, смартфони), мережеві пристрої (маршрутизатори, комутатори) та сенсорні периферійні пристрої (тактильні екрани, кнопки з підвищеною чутливістю тощо).

Блок *Програмне забезпечення* поділяється на три підблоки: інтерфейс користувача (UI Layer), логіка додатку (Business Logic Layer), доступ до даних (Data Access Layer).

Блок *Середовище користувача* включає підмодулі: користувач з інклюзією, адаптивні налаштування, сценарії використання, допоміжні засоби (екранні диктори, голосові команди, збільшувачі екрану).

Зв'язки між цими блоками позначають потоки управління, даних та подій. Наприклад, клієнтські пристрої взаємодіють з інтерфейсом користувача через стандартні порти передачі даних, а також через спеціальні модулі адаптації.

Діаграма внутрішніх блоків: "Компоненти програмного забезпечення системи" у Додатку А на малюнку А.11. Ця діаграма деталізує внутрішню структуру програмного блоку, де представлено зв'язки між модулями через порти передачі повідомлень. Основні частини:

- UI-компонент (форма входу, панель налаштувань, головна панель доступу);
- Модуль адаптації (базовий адаптер, профілюючий модуль, шаблони інтерфейсу);
- Модуль управління сеансами (менеджер сесій, контролер автентифікації);

- Модуль доступу до бази даних (ORM-контролер, фільтри безпеки, кеш-зв'язок);
- Бібліотеки доступності (екранний диктор, голосові помічники, сенсорна навігація).

Усі компоненти мають вхідні та вихідні порти. Наприклад, UI-компонент має порти: вхід (отримання профілю) і вихід (ініціація адаптації). Модуль адаптації отримує дані з профілю користувача, обробляє їх і через порт подає готовий шаблон до UI.

У структурній моделі UML використано чіткий поділ на пакети:

- Entity (класи-сутності): *UserProfile*, *AccessibilityPreferences*, *UsageStatistics*, *SessionLog*;
- Boundary (граничні класи): *LoginForm*, *SettingsPanel*, *MainUI*, *FeedbackForm*;
- Control (класи-керування): *AdaptationManager*, *SessionController*, *FeedbackProcessor*, *DataAnalyzer*.

Пакет *Entity* зосереджує класи, що відповідають за збереження стану і довготривале управління даними. Наприклад, клас *UserProfile* містить атрибути: *userID*, *name*, *disabilityType*, *language*, *theme*, *fontSize*. Його методи: *updateProfile()*, *applyDefaults()*, *savePreferences()*.

Пакет *Boundary* містить класи, які реалізують інтерфейсну взаємодію. *LoginForm* має атрибути *login*, *password*, *errorMessages* та методи *validateInput()*, *submitForm()*. *MainUI* – основне вікно програми, яке динамічно перебудовується в залежності від адаптацій.

Класи *Control* відповідають за логіку керування. Наприклад, *AdaptationManager* обробляє запити на зміну інтерфейсу, аналізує профіль, обирає відповідний шаблон. Має методи: *loadProfileData()*, *generateAdaptation()*, *applyInterfaceTemplate()*.

Діаграма класів, що наведена у Додатку А на малюнку А.12 представляє архітектуру системи, що включає об'єкти для управління користувачами, товарами, кошиком, доставкою, платежами та взаємодією з супермаркетами.

Вона демонструє взаємозв'язки між основними сутностями та компонентами системи, такими як UI-сторінки, контролери, сервіси та репозиторії.

Опис ключових класів та їх взаємозв'язків:

1. Користувач (User):

- *Атрибути:* id, adress, firstName, idString, lastName, number, phone, email, password.
- *Методи:* login(), register(), updateAdress(), updatePhoneNumb().
- *Взаємозв'язки:*
 - Асоціація з Cart (1 до 1..1) – один користувач може мати один кошик.
 - Асоціація з Delivery (1 до 1..1) – один користувач може мати одну доставку.
 - Асоціація з Selection Interface (1 до 1..1) – один користувач може мати один інтерфейс вибору.
- *Компоненти взаємодії:* LoginPage (UI), AuthController (контролер), UserRepository (репозиторій). Ці компоненти керують автентифікацією та даними користувача.

2. Кошик (Cart)

- **Атрибути:** id, product_id, quantity, user_id, sum.
- **Методи:** addProductToCart(), removeitemFromCart(), updateQuantity().
- **Взаємозв'язки:**
 - Асоціація з User (1 до 1..1) – один кошик належить одному користувачеві.
 - Асоціація з Product (1 до N) – один кошик може містити багато товарів.

Компоненти взаємодії: CartPage (UI), CartController (контролер), CartService (сервіс), CartRepository (репозиторій). Ці компоненти керують логікою кошика.

3. Товар (Product)

- **Атрибути:** id, description, category, image, data, name, price, quantity, supermarket_id.

- **Методи:** `getDetails()`, `updatePrice()`, `updateStockQuantity()`.
- **Взаємозв'язки:**
 - Асоціація з `Cart` (N до 1) – багато товарів можуть бути в одному кошику.
 - Асоціація з `Category` (1 до N) – один товар належить до однієї категорії.
 - Асоціація з `Supermarket` (N до 1) – багато товарів можуть належати до одного супермаркету.
- **Компоненти взаємодії:** `ProductPage` (UI), `ProductController` (контролер), `ProductRepository` (репозиторій). Ці компоненти керують даними про товари.

4. Категорія (Category)

- **Атрибути:** `id`, `name`, `string`.
- **Методи:** `getCategoryName()`, `renameCategoryName()`.
- **Взаємозв'язки:**
 - Асоціація з `Product` (1 до N) – одна категорія може мати багато товарів.
- **Компоненти взаємодії:** `CategoryPage` (UI), `CategoryController` (контролер), `CategoryRepository` (репозиторій). Ці компоненти керують даними про категорії.

5. Супермаркет (Supermarket)

- **Атрибути:** `id`, `name`, `string`.
- **Методи:** `getSupermarketName()`, `renameSupermarket()`.
- **Взаємозв'язки:**
 - Асоціація з `Product` (1 до N) – один супермаркет може мати багато товарів.
- **Компоненти взаємодії:** `SupermarketPage` (UI), `SupermarketController` (контролер), `SupermarketRepository` (репозиторій). Ці компоненти керують даними про супермаркети.

6. Доставка (Delivery)

– **Атрибути:** `client_adress`, `delivery_method`, `product_id`, `int`, `user_id`, `string`, `delivery_id`, `price`.

– **Методи:** `changeDeliveryMethod()`, `scheduleDelivery()`, `updateAddressData()`.

– **Взаємозв'язки:**

- Асоціація з `User` (1 до 1..1) – одна доставка належить одному користувачеві.

– **Компоненти взаємодії:** `DeliveryPage` (UI), `DeliveryController` (контролер), `DeliveryService` (сервіс), `DeliveryRepository` (репозиторій). Ці компоненти керують процесами доставки.

7. Оплата (Payment)

– **Атрибути:** `id`, `delivery_id`, `payment_method`, `choice`.

– **Методи:** `changeMethod()`.

– **Взаємозв'язки:**

- Асоціація з `Delivery` (1 до 1) – одна оплата пов'язана з однією доставкою.

– **Компоненти взаємодії:** `PaymentPage` (UI), `PaymentController` (контролер), `PaymentService` (сервіс), `PaymentRepository` (репозиторій). Ці компоненти керують платіжними операціями.

8. Інтерфейс вибору (Selection Interface)

– **Атрибути:** `description`, `string`, `id`, `name`, `delivery_method`, `user_id`, `string`.

– **Методи:** `selectOption()`, `updateDiscription()`.

– **Взаємозв'язки:**

- Асоціація з `User` (1 до 1..1) – один інтерфейс вибору належить одному користувачеві.

– **Компоненти взаємодії:** `SelectionPage` (UI), `SelectionInterfaceController` (контролер), `SelectionService` (сервіс), `SelectionRepository` (репозиторій). Ці компоненти керують вибором опцій.

Загальна структура та компоненти: Діаграма використовує архітектурний шаблон, де:

...Page (наприклад, LoginPage, CartPage) представляють UI-шари, які взаємодіють з користувачем.

...Controller (наприклад, AuthController, CartController) обробляють запити від UI та координують взаємодію між сервісами та репозиторіями.

...Service (наприклад, PaymentService, CartService) містять бізнес-логіку та виконують операції, такі як додавання до кошика, розрахунок цін.

...Repository (наприклад, UserRepository, ProductRepository) надають абстракцію для доступу до даних, взаємодіючи з базою даних або іншим сховищем даних.

Пунктирні стрілки вказують на залежності або використання одних компонентів іншими, відображаючи потік управління та даних в системі. Ця архітектура сприяє розділенню відповідальностей, що робить систему більш модульною, масштабованою та легкою для підтримки. Модель чітко розмежовує обов'язки між даними, логікою й інтерфейсом, що відповідає принципам MVC-архітектури. Інтеграція допоміжних технологій (екранні диктори, голосові команди) реалізована як окремі класи, що дозволяє легко масштабувати систему без порушення основної структури.

Завдяки використанню структурних діаграм SysML і UML було досягнуто високого рівня формалізації та прозорості проєкту, що є критично важливим на етапах тестування, впровадження та підтримки програмного продукту. Структурне моделювання забезпечує відповідність архітектури системи функціональним і нефункціональним вимогам, викладеним у попередніх розділах.

РОЗДІЛ 3 ПРОЄКТУВАННЯ І РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Інформаційне забезпечення

3.1.1 Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення розробленої інформаційної системи з адаптивним інтерфейсом для осіб з інклюзією включає сукупність даних, які забезпечують коректну взаємодію користувача з інтерфейсом, налаштування адаптивних режимів, формування замовлення та підтримку повного життєвого циклу користувацької сесії. Інформаційне наповнення визначається виходячи з функціональних вимог до системи та орієнтоване на простоту, доступність і персоналізацію.

До основних інформаційних об'єктів системи належать:

- персональні дані користувача (ім'я, email, пароль, тип інклюзії);
- параметри адаптивного режиму (наприклад, високий контраст, озвучення, збільшені елементи);
- перелік супермаркетів із логотипами;
- каталог товарів із назвами, зображеннями, цінами, категоріями;
- вміст кошика (товари, кількість, ціна, загальна сума);
- дані для оформлення замовлення (адреса, спосіб доставки, метод оплати).

Інформація в системі структурована у вигляді взаємопов'язаних сутностей (рис. 3.1). Це дозволяє не лише організувати інтерфейс логічно, але й реалізувати адаптивну логіку змін відображення даних відповідно до особливостей користувача. На зображенні представлено логічну схему інформаційних об'єктів, які формують ядро користувацької взаємодії із системою. Зліва направо демонструється послідовність даних: від користувача, який обирає режим доступності, до оформлення замовлення. Кожен блок відповідає окремій групі інформації, яка реалізується в інтерфейсі

прототипу. Стрілки позначають напрямок передачі даних та залежності між об'єктами.

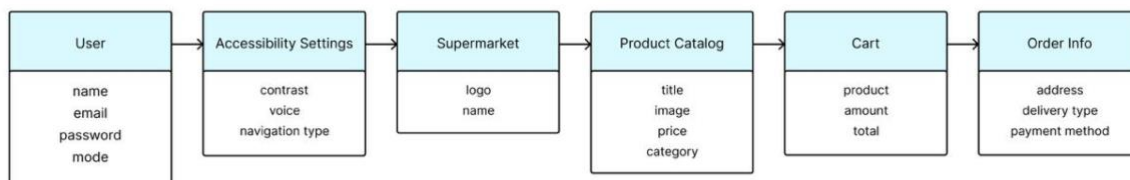


Рисунок 3.1 – Візуалізація структури інформаційного забезпечення системи

Джерело: розроблено автором на основі виконаного дослідження

Особливу увагу приділено доступності інформації: вся текстова інформація має бути читабельною для екранних зчитувачів, елементи керування — достатньо контрастними, а всі ключові дії — озвучуваними або дубльованими візуально. Це забезпечує відповідність сучасним стандартам інклюзивного дизайну, таким як WCAG 2.1, і робить інформацію доступною для найширшого кола користувачів.

3.1.2 Категорії даних, що відображаються в інтерфейсі

Одним із ключових елементів інформаційного забезпечення системи є організація категорій даних, які безпосередньо відображаються в інтерфейсі користувача. Всі дані згруповані таким чином, щоб забезпечити інтуїтивно зрозумілу структуру, адаптивність для різних типів інклюзії, а також спростити навігацію мобільним застосунком. Структура категорій розроблена відповідно до функціональних вимог та UX-практик для доступних інтерфейсів.

Категорія 1. Дані користувача

Ці дані містяться у формі реєстрації/входу та профілі користувача. До них належать:

- ім'я користувача;
- адреса електронної пошти;
- пароль (збережений зашифровано);
- обраний режим доступності (visual, motor, cognitive);
- історія попередніх взаємодій (у майбутній реалізації — для персоналізації).

Ці дані є основою для створення сесії користувача та подальшої адаптації інтерфейсу згідно з його особливостями.

Категорія 2. Список супермаркетів

Інформація про доступні торгові мережі подається у вигляді інтерактивних карток з логотипом, назвою магазину та кольоровим маркуванням. Всі супермаркети виведені у єдиному стилі, з дотриманням принципів контрастності та читабельності. Візуальні елементи легко розпізнаються користувачами з порушеннями зору або когнітивним сприйняттям.

Категорія 3. Каталог товарів

Основний функціональний блок інтерфейсу, який виводить перелік продуктів у вигляді сітки або списку. Дані, які відображаються для кожного товару:

- назва;
- фотографія;
- ціна (одинична);
- категорія товару;
- кнопка «додати до кошика».

Інтерфейс підтримує фільтрацію за категоріями, до яких належать:

- овочі, фрукти та зелень;
- молочні продукти;
- бакалія;

- напої;
- хлібобулочні вироби;
- заморожені продукти;
- побутова хімія;
- інше.

Категорія 4. Кошик

Інформація, що відображається в кошику, включає:

- список обраних товарів;
- кількість кожного товару;
- ціна за одиницю;
- загальна вартість;
- кнопки для редагування кількості або видалення позиції.

Система динамічно оновлює ці значення, підтримуючи адаптивне відображення: збільшені кнопки, можливість озвучення товару при наведенні, спрощене сортування.

Категорія 5. Дані доставки

Після підтвердження кошика користувач переходить на етап введення даних доставки:

- адреса отримувача;
- вибір типу доставки: кур'єр або самовивіз;
- вибір дати та часу.

Усі поля мають візуальну та аудіо-підтримку, що дозволяє людям з інвалідністю комфортно вводити інформацію.

Категорія 6. Оплата

Останній блок містить дані щодо методу оплати:

- готівка;
- банківська картка;
- онлайн-переказ (у майбутній реалізації).

Інтерфейс забезпечує однозначність вибору, а для користувачів з моторними порушеннями — збільшені інтерактивні елементи.

У підсумку кожна категорія інформації адаптована під вимоги доступності, що дозволяє різним користувачам виконувати дії без додаткової допомоги.

3.1.3 Способи отримання та оновлення даних в інтерфейсі (імітація в прототипі)

Оскільки в межах даної кваліфікаційної роботи було створено саме прототип, а не програмну реалізацію системи, процеси обміну даними між користувачем і системою виконуються в імітованому вигляді. Прототип, реалізований у середовищі Figma, моделює логіку оновлення інформації, яка мала б відбуватися в повноцінній інформаційній системі з підключенням до бази даних та серверної логіки.

Основні підходи до імітації оновлення даних у Figma:

1. Інтерактивні переходи (*interactive prototyping*)

Усі ключові дії користувача пов'язані з переходом між окремими екранами (фреймами). Наприклад, після натискання кнопки «Start» здійснюється перехід до форми реєстрації, а після успішної реєстрації – до екрана вибору режиму доступності.



Рисунок 3.2 – Інтерактивні переходи

Джерело: розроблено автором на основі виконаного дослідження

2. Component variants (варіанти компонентів)

Для імітації станів (додано товар до кошика, змінено кількість, вибрано спосіб доставки) використано компоненти з варіантами. Це дозволяє змінювати візуальний вигляд елементів залежно від дій користувача — без коду, але з реалістичною поведінкою.

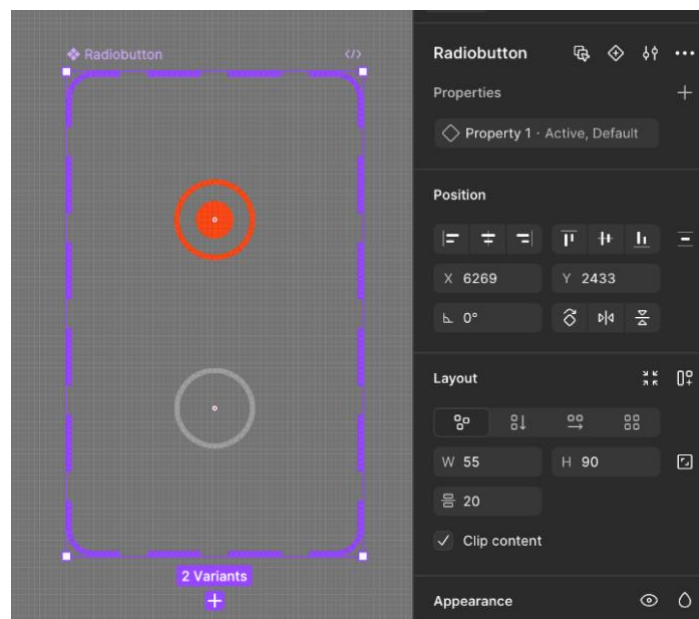


Рисунок 3.3 – Приклад варіанту компоненту радіо-кнопки (активний/обраний та пасивний/не обраний стани)

Джерело: розроблено автором на основі виконаного дослідження

3. Імітація динамічної інформації

Дані, які у реальному додатку надходили б через API (наприклад, каталог товарів, список у кошику), у Figma представлені статично, але логічно:

- картки товарів змінюються при переході за категоріями;
- кошик оновлюється вручну, натисканням на кнопки «+»/«-»;
- дані замовлення передаються умовно при переході між екранами.

4. Створення ілюзії стану системи

Для кожного сценарію (наприклад, додавання товару, оформлення замовлення, підтвердження) створено окремий екран або варіант фрейму, який

демонструє результат дії. Це дозволяє відтворити візуальну послідовність обробки даних.

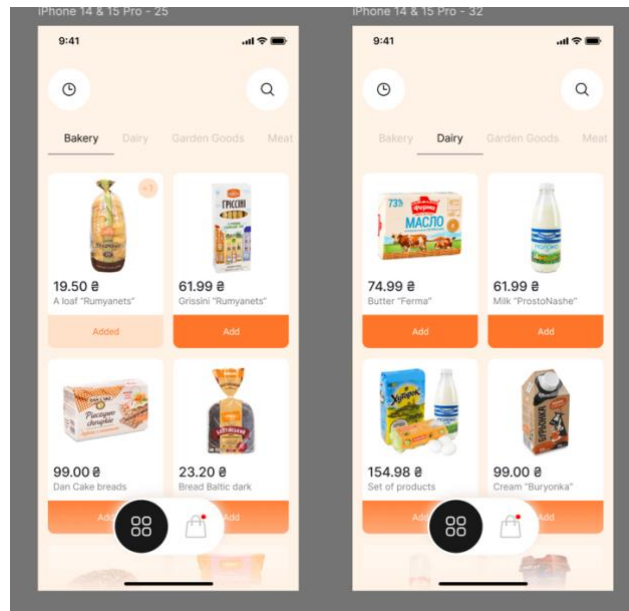


Рисунок 3.4 – Динамічна інформація та стан системи (додавання до кошику, зміна категорії товарів)

Джерело: розроблено автором на основі виконаного дослідження

5. Умовне збереження та озвучення

Хоча прототип не взаємодіє з голосовими сервісами, кнопки, що належать до режиму озвучення, мають окреме візуальне оформлення (наприклад, іконка гучномовця), що демонструє логіку майбутнього озвучення при активованому режимі доступності.

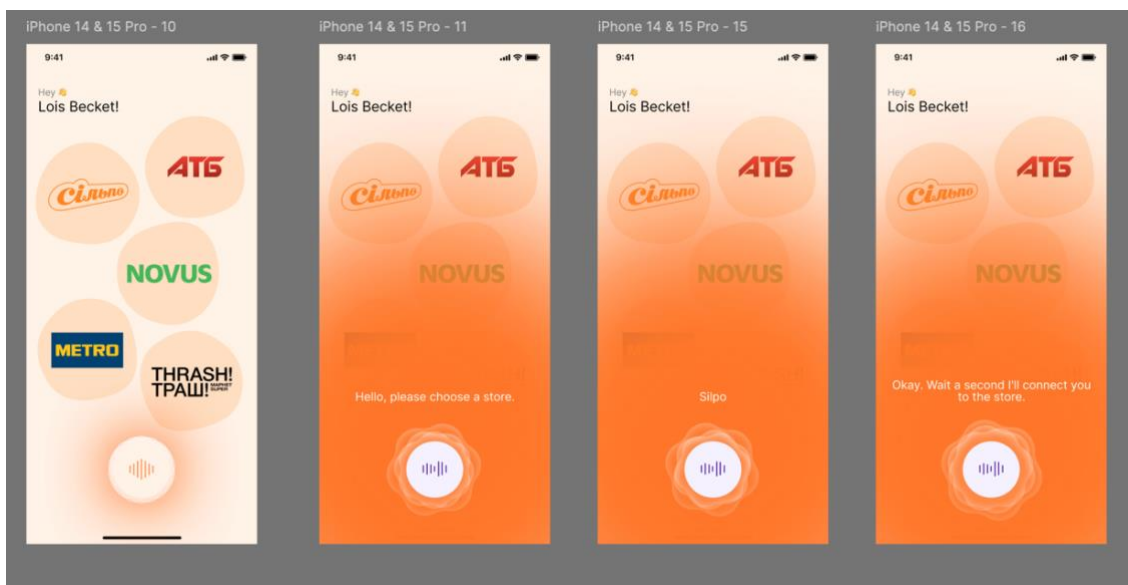


Рисунок 3.5 – Приклад анімації голосового введення

Джерело: розроблено автором на основі виконаного дослідження

Приклади імітації оновлення даних:

Додавання товару до кошика. Плитка товару змінюється на іншу з виведеним числом (кількість), після натискання «+». Це відображає зміну стану товару в кошику.

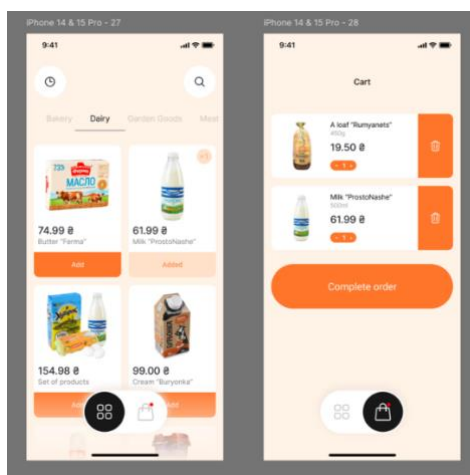


Рисунок 3.6 – Додавання товару до кошика

Джерело: розроблено автором на основі виконаного дослідження

Редагування доставки. Після введення адреси вручну (на прототипі – у вигляді текстового поля), перехід до наступного екрану відтворює логіку збереження введеної інформації.

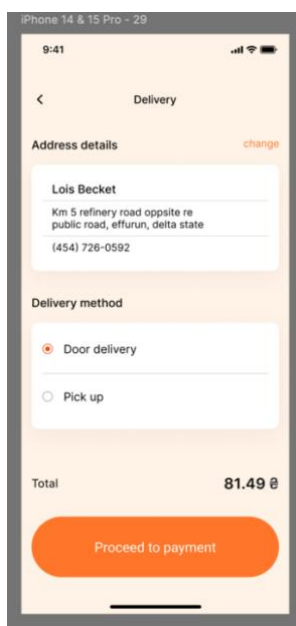


Рис. 3.7 – Редагування доставки

Джерело: розроблено автором на основі виконаного дослідження

Підтвердження замовлення. Виводиться окремий екран з повідомленням «Success» і кнопкою «Thank you», що імітує логіку зворотного зв'язку від системи.

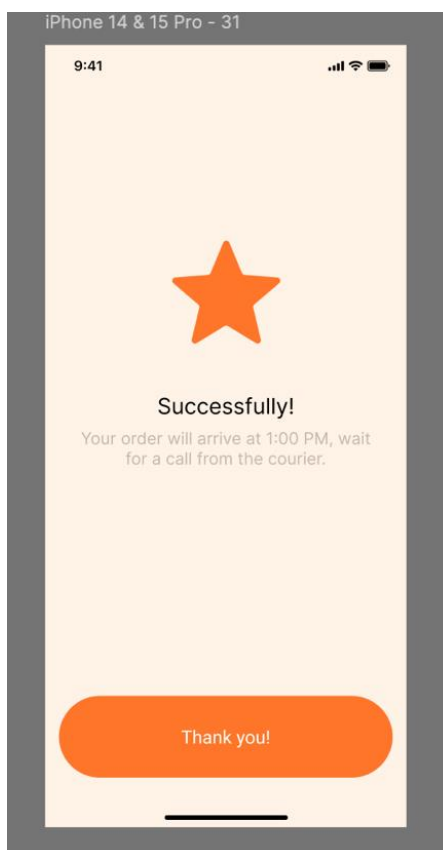


Рисунок 3.8 – Імітація підтвердження замовлення

Джерело: розроблено автором на основі виконаного дослідження

Логіка, яку в майбутньому реалізуватимуть програмно:

У прототипі реалізовано імітацію, проте в реальному середовищі ці дії будуть відповідати таким запитам:

- GET-запит до бази даних для отримання товарів;
- POST-запит для створення користувача або оформлення замовлення;
- PUT-запит для оновлення кількості товару в кошику;
- PATCH або DELETE — для редагування / видалення елементів.

Прототип у Figma таким чином виконує не лише дизайнерську функцію, а й імітує поведінку справжньої системи, що дозволяє тестувати логіку навігації та перевіряти зручність користувацького досвіду ще до початку програмної реалізації.

3.2 Прототипування користувацького інтерфейсу

3.2.1 Інструментальні засоби (Figma)

Основним інструментом для створення та реалізації прототипу адаптивного інтерфейсу в межах даного проекту було обрано сервіс Figma. Це сучасне веборієнтоване середовище для дизайну інтерфейсів, що дозволяє створювати інтерактивні прототипи без написання коду, з можливістю адаптації до різних форматів пристроїв і врахування потреб користувачів з інклюзією.

Figma була обрана з огляду на низку переваг:

- підтримка хмарної роботи без потреби в установці ПЗ;
- зручна реалізація інтерактивності (переходи, стани, кнопки);
- система компонентів і варіантів, яка дозволяє масштабувати інтерфейс і впроваджувати адаптивність;
- можливість створення адаптивного дизайну під мобільні пристрої;

- підтримка спільної роботи (collaborative design), що особливо важливо для командної розробки.

У процесі розробки прототипу в Figma було створено понад 10 екранів, які охоплюють повний користувацький сценарій — від початкового входу до оформлення замовлення та підтвердження. Крім стандартної версії інтерфейсу, було реалізовано адаптивні варіанти для трьох типів користувачів:

- осіб із порушенням зору (високий контраст, великі шрифти);
- осіб із порушеннями моторики (збільшені кнопки, мінімізація жестових дій);
- осіб із когнітивними особливостями (спрощена навігація, мінімум візуальних відволікачів, підказки).

Засоби Figma дозволили швидко протестувати візуальну логіку, визначити вузькі місця у навігації, а також підготувати прототип до майбутньої реалізації у вигляді коду. Використання цього інструменту надало змогу створити багатошарові екрани з реалістичною поведінкою, візуалізувати усі функціональні вимоги системи та забезпечити наочність для юзабіліті-тестування з потенційними користувачами.

Таким чином, Figma виступила не просто як дизайнерський засіб, а як ключовий етап технічного моделювання, що дозволив протестувати і верифікувати концепцію адаптивного інтерфейсу до початку фактичної розробки.

3.2.2 Етапи створення прототипу

Створення прототипу інформаційної системи з адаптивним інтерфейсом є ключовим етапом візуального моделювання продукту. Прототипування дозволяє сформулювати уявлення про логіку роботи системи, перевірити зручність інтерфейсу, а також забезпечити тестування функціональних сценаріїв до початку програмної реалізації. У межах даної роботи процес проектування в Figma проходив послідовно, охоплюючи всі основні етапи UI/UX-розробки.

Етап 1. Визначення користувацьких сценаріїв

Першим кроком стало формування ключових сценаріїв використання системи: від входу в застосунок і реєстрації, до вибору товарів, оформлення замовлення та оплати. Особливу увагу було приділено адаптації під користувачів із різними типами інклюзії, тому кожен сценарій проектувався із урахуванням доступності.

Етап 2. Створення wireframes (каркасів інтерфейсу)

На цьому етапі були побудовані низькорівневі каркасні макети (wireframes). Вони включали лише базову структуру інтерфейсу: розташування кнопок, полів введення, зон контенту. Завдяки цьому вдалося визначити оптимальну логіку розміщення елементів та оцінити загальну простоту навігації.

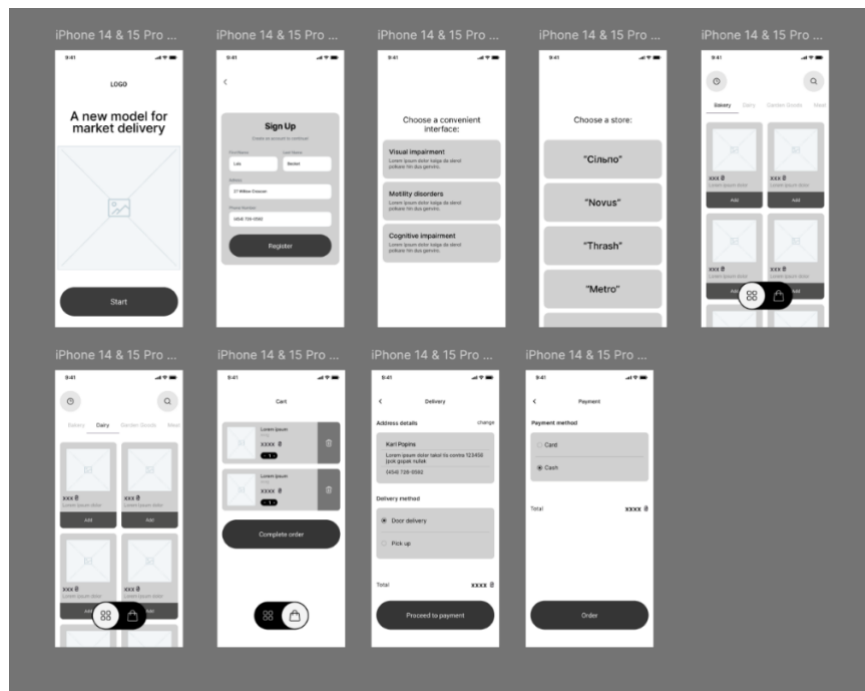


Рисунок 3.9 – Редагування доставки

Джерело: розроблено автором на основі виконаного дослідження

Етап 3. Опрацювання стилістики та UI-компонентів

Наступним кроком стало застосування фірмового стилю, підбір кольорів, типографіки, розмірів кнопок та шрифтів. Було створено бібліотеку компонентів: кнопки, товарні плитки, форми тощо. З метою адаптації до користувачів з інвалідністю, були передбачені варіанти з високим контрастом, збільшеними шрифтами, спрощеною графікою.

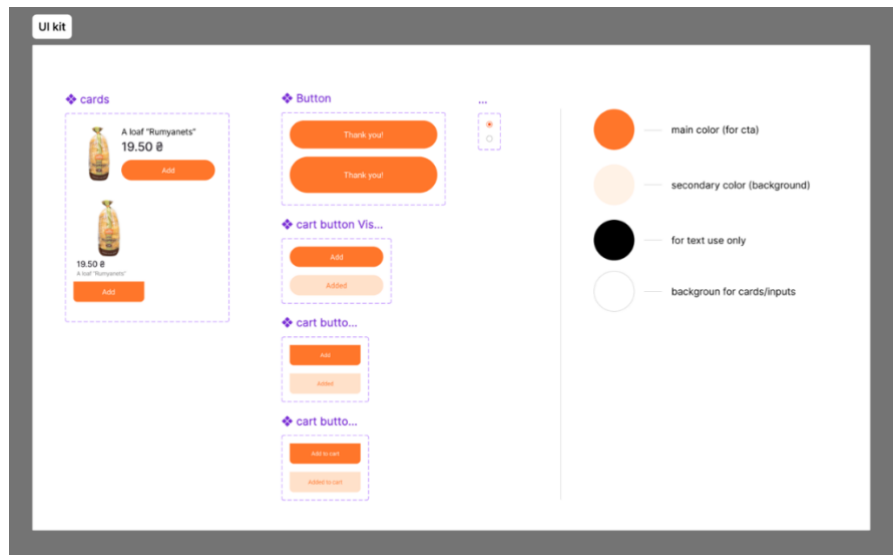


Рисунок 3.10 – UI kit (бібліотека компонентів та кольорів)

Джерело: розроблено автором на основі виконаного дослідження

Етап 4. Створення основних екранів

На основі UI-компонентів було створено набір екранів, які відображають усі основні функції системи:

- стартовий екран;
- форма реєстрації;
- вибір режиму доступності;
- перелік супермаркетів;
- каталог товарів;
- кошик;
- доставка й оплата;
- повідомлення про успішне замовлення.

Усі екрани проєктувались з урахуванням UX-принципів: мінімалізм, ієрархія, логічна послідовність дій.

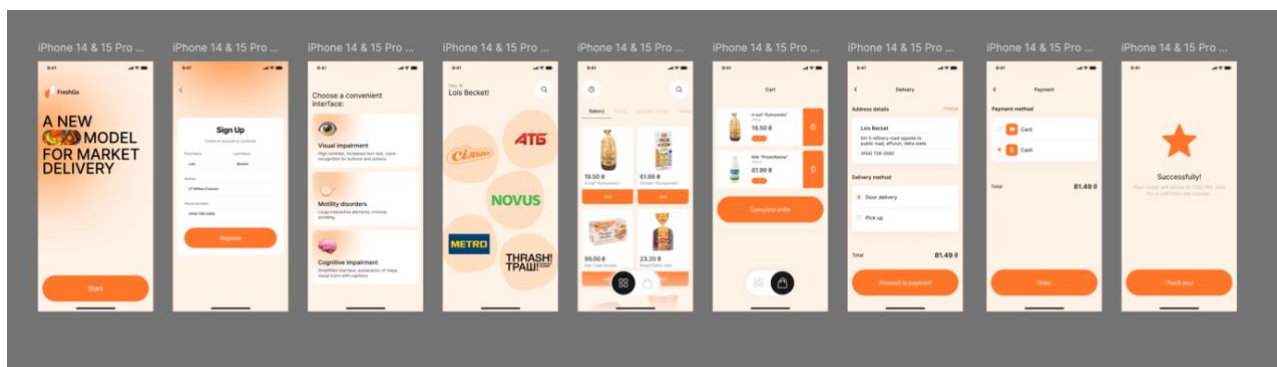


Рисунок 3.11 – Набір основних екранів

Джерело: розроблено автором на основі виконаного дослідження

Етап 5. Додавання адаптивних режимів

Було створено окремі варіанти інтерфейсу для кожного типу інклюзії:

- режим зорової доступності (високий контраст, великий шрифт, голосова підтримка);
- режим моторної доступності (збільшені кнопки, інтерфейс без жестикуляцій);
- режим когнітивної доступності (спрощена структура, додаткові підказки, відсутність відволікаючих елементів).

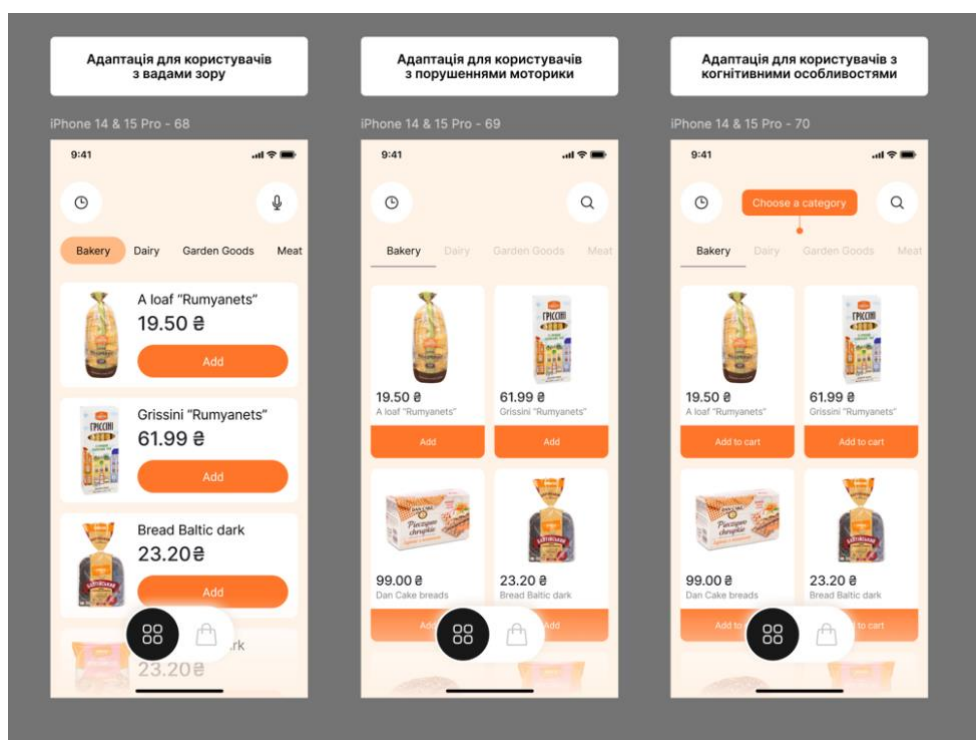


Рисунок 3.12 – Приклад адаптації під різні типи інклюзії

Джерело: розроблено автором на основі виконаного дослідження

Етап 6. Інтерактивне прототипування

На завершальному етапі всі екрани були пов'язані між собою за допомогою інтерактивних переходів (див.рис.3.1.2). Це дало змогу реалізувати умовну поведінку системи:

- перехід після натискання на кнопки;
- оновлення стану товарів у кошику;
- умовне відображення повідомлень тощо.

Інтерактивність дозволяє повністю пройти всі користувацькі сценарії — від входу до підтвердження замовлення — навіть без підключення до коду чи бази даних. Таким чином, прототип у Figma виконує роль візуального MVP (мінімально життєздатного продукту).

3.2.3 Принципи UI/UX-дизайну, застосовані при прототипуванні

У процесі створення прототипу особливу увагу було приділено дотриманню сучасних принципів UI/UX-дизайну, що забезпечують зручність, логічність та доступність інтерфейсу. З урахуванням специфіки системи, орієнтованої на користувачів з інклюзією, дизайн інтерфейсу було побудовано з дотриманням як стандартних UX-рекомендацій, так і спеціалізованих вимог до доступності (accessibility).

Мінімалізм та простота

Інтерфейс має максимально лаконічну структуру. Уникнення перевантаження візуальними елементами, зменшення кількості тексту, продумана ієрархія компонентів дозволяють користувачам швидко орієнтуватися в інтерфейсі, навіть за наявності когнітивних або візуальних особливостей.

Ієрархічність та логічна структура

Кожен екран прототипу має чітку логіку: користувач бачить лише ті елементи, які потрібні на поточному етапі взаємодії. Кнопки розміщені у передбачуваних місцях, що зменшує когнітивне навантаження.

Стійка навігація

Система має зрозумілу навігаційну структуру: всі переходи між екранами виконуються однозначно, без зайвих кроків. У нижній частині інтерфейсу реалізована панель швидкого доступу до ключових розділів: каталог, кошик.

Адаптивність до потреб користувачів

Під час створення прототипу було дотримано принципів інклюзивного дизайну:

- використано шрифти достатнього розміру (не менше 16 pt у звичайному режимі, масштабовані у режимах доступності);
- передбачено високий контраст кольорів для користувачів з порушеннями зору;
- кнопки збільшено та розміщено на безпечній відстані одна від одної для користувачів із моторними порушеннями;
- система не містить анімацій, що могли б відволікати або викликати дезорієнтацію.

Консистентність елементів

Використання спільних компонентів і стилів на всіх екранах забезпечує єдність візуальної мови інтерфейсу. Це знижує ризик помилок при взаємодії та підвищує загальну інтуїтивність використання.

Візуальний фокус і керування увагою

На кожному етапі користувач бачить лише релевантні дії. Сторонні елементи приховано або винесено на другий план, що дозволяє сконцентруватися на поточному завданні.

У результаті застосування наведених принципів вдалося створити адаптивний інтерфейс, який одночасно є сучасним, інтуїтивно зрозумілим та доступним для людей з різними формами інклюзії.

3.2.4 Адаптація під потреби користувачів з інклюзією

В основі проекту закладено принцип адаптивності інтерфейсу — це означає, що система здатна враховувати індивідуальні потреби користувача

вже з моменту першої взаємодії. У створеному прототипі реалізовано можливість вибору одного з трьох спеціалізованих режимів доступності, кожен з яких відповідає певному типу порушень: зору, моторики або когнітивного сприйняття. Після активації режиму інтерфейс змінюється відповідним чином, забезпечуючи комфортне користування системою.

На відміну від темних інтерфейсів, які зазвичай асоціюються з високою контрастністю, обраний дизайн у Figma має світле, чисте оформлення, де адаптація контрасту реалізується через збільшення яскравості елементів, акцентів і чіткого візуального поділу між зонами. Це дозволяє досягти доступності без застосування агресивних темних тонів, що є важливим для збереження легкого сприйняття і водночас відповідності стандартам WCAG.

Адаптація для користувачів з порушеннями зору

У цьому режимі інтерфейс змінюється з урахуванням потреб людей із частковою втратою зору, зниженим контрастним сприйняттям або іншими візуальними труднощами. У прототипі реалізовано:

- підвищену контрастність елементів інтерфейсу — кнопки, поля, тексти мають насичене окреслення, темно-сині або чорні написи на білому фоні, акцентні кольори використовуються для підсвічування активних елементів;
- збільшені шрифти заголовків і кнопок, щоб їх легко було зчитувати навіть без масштабування пристрою;
- логічну візуальну ієрархію — важлива інформація виділена жирним або кольором, а допоміжні елементи — світліші;
- озвучення ключових елементів передбачене як функція, що буде реалізована у повній версії. Прототип демонструє її наявність через іконки мікрофона/озвучення, що візуально підказують користувачу про можливість голосового супроводу.

Адаптація для користувачів з порушеннями моторики

Користувачі з порушенням дрібної моторики мають труднощі з точністю торкання або натискання на дрібні елементи. Саме тому в прототипі передбачено:

- збільшені розміри кнопок та інтерфейсних елементів, які займають більше площі екрана й мають великі зони натискання;
- розташування кнопок на безпечній відстані одна від одної, щоб уникнути випадкових натискань;
- мінімізацію кількості дій — на кожному екрані передбачено найнеобхідніші функції, без потреби перемикатись між численними вкладками;
- навігацію великими інтерактивними картками (наприклад, вибір магазину або категорії товарів), що легко сприймаються і зручно натискаються пальцем.

Адаптація для користувачів з когнітивними особливостями

Користувачі з когнітивними порушеннями потребують простого й передбачуваного інтерфейсу без складної логіки. У Figma це реалізовано через:

- лінійну, покрокову навігацію — усі дії користувача логічно впорядковані: після реєстрації → режим доступності → вибір магазину → каталог → кошик → доставка → оплата;
- мінімум тексту — формулювання короткі, чіткі, без надлишкових інструкцій;
- іконки й візуальні підказки — усі дії супроводжуються знайомими символами (кошик, плюс, стрілка назад), що спрощують розуміння;
- відсутність відволікаючих анімацій або складних елементів — екрани не містять "візуального шуму", що допомагає сфокусувати увагу.

У підсумку реалізація адаптації в прототипі базується на змінюванні структури, розмірів, контрастності та навігаційної логіки в залежності від обраного режиму доступності. Такий підхід дозволяє ще на стадії проєктування перевірити, наскільки майбутня система буде зручною, інклюзивною та дружньою до різних груп користувачів.

3.3 Візуалізація ключових екранів

3.3.1 Початковий екран та реєстрація

Початковий екран у розробленому прототипі виконує функцію вітального інтерфейсу, який знайомить користувача з застосунком і пропонує перейти до подальших дій. Він має лаконічну та інтуїтивно зрозумілу структуру, що відповідає принципам доступності й UX-дизайну.

У верхній частині екрана розміщено логотип застосунку та короткий слоган, який описує призначення сервісу (наприклад, «Your groceries, your way»). У центрі екрана розташовано вітальний текст, який звертається до користувача, створюючи відчуття персоналізованої взаємодії.

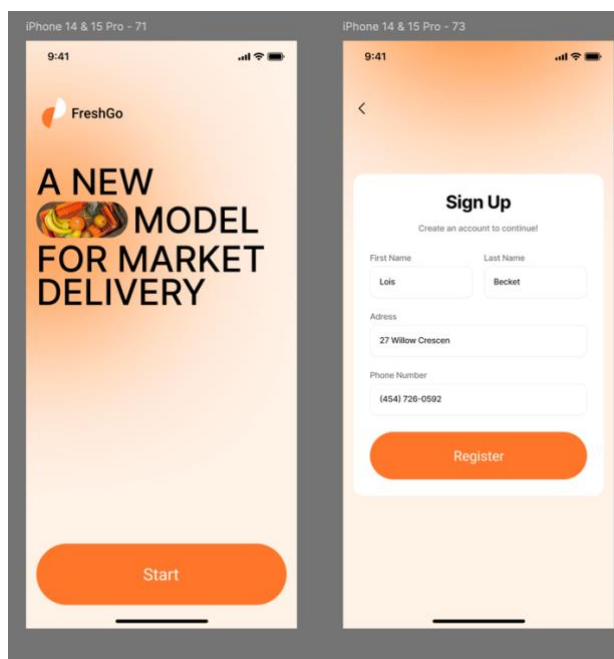


Рисунок 3.13 – Початковий екран та екран авторизації користувача

Джерело: розроблено автором на основі виконаного дослідження

Ключовим елементом цього екрана є кнопка “Start”, яка відкриває форму реєстрації. Кнопка має великий розмір, контрастний колір та розміщується по центру екрана — це забезпечує її зручне використання користувачами з порушеннями моторики та зору.

Після натискання кнопки “Start” система переходить до екрана реєстрації, де користувачеві пропонується ввести три основні поля:

- ім'я;
- електронну адресу (email);
- пароль.

Інтерфейс форми реєстрації також відповідає принципам адаптації:

- поля мають чіткі рамки й достатній відступ;
- елементи розташовані вертикально, щоб уникнути плутанини.

Після введення даних і натискання кнопки “Register”, користувач автоматично переходить до наступного етапу — вибору режиму доступності. Таким чином, реєстрація поєднана з налаштуванням адаптації, що знижує кількість кроків для користувача та дозволяє одразу підлаштувати інтерфейс під його потреби.

Ці екрани реалізовано у Figma з інтерактивними зв'язками між фреймами, що дозволяє в повній мірі імітувати логіку реального застосунку ще до початку кодування.

3.3.1 Вибір режиму доступності

Після проходження реєстрації користувач автоматично переходить на екран вибору режиму доступності, що є ключовим елементом адаптивної логіки розробленого інтерфейсу. Саме на цьому етапі система запитує в користувача, яким чином слід адаптувати інтерфейс з урахуванням його індивідуальних потреб.

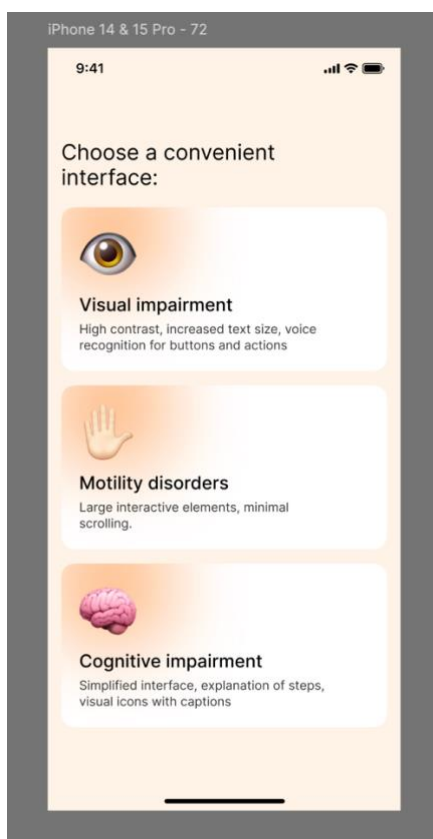


Рисунок 3.14 – Екран вибору режиму доступності

Джерело: розроблено автором на основі виконаного дослідження

Екран оформлено у зрозумілому візуальному стилі: назва кожного режиму супроводжується ілюстративною іконкою, коротким описом функціональності та кнопкою вибору. Це дозволяє навіть недосвідченому користувачу легко зорієнтуватися й обрати відповідний варіант.

Усього реалізовано три основні режими:

- Visual accessibility – для користувачів із порушенням зору. Включає: збільшений розмір шрифтів, контрастні кольори, великі кнопки, можливість подальшого озвучення.
- Motor accessibility – для користувачів із порушеннями моторики. Включає: великі зони натискання, зменшену кількість елементів на екрані, спрощену навігацію без складних жестів.
- Cognitive accessibility – для користувачів із когнітивними особливостями. Включає: спрощений текст, чітку послідовність дій, підказки, стабільне розташування кнопок і відсутність відволікаючих елементів.

Дизайн цього екрана відповідає ключовим вимогам UX:

- мінімум дій — користувач обирає режим одним кліком;
- візуальна зрозумілість — кожен варіант має не лише назву, а й короткий опис дії (наприклад, “для слабкого зору”);
- підказка через іконки, що спрощує розуміння;
- інтуїтивне розміщення — елементи вибору мають однаковий розмір, відступи та логічну вертикальну або сіткову структуру.

Після обрання режиму доступності система переходить до основного функціоналу вже з активованими налаштуваннями, що визначають вигляд подальших екранів (контраст, розміри, структура). Це дозволяє забезпечити безперервний досвід користування для людей з інклюзією.

3.3.3 Каталог товарів, кошик, оформлення замовлення

Центральною частиною функціоналу розробленої інформаційної системи є взаємодія користувача з каталогом товарів, формуванням замовлення та його подальшим оформленням. У прототипі ці процеси реалізовано у вигляді послідовності логічно пов’язаних між собою екранів, що відповідають ключовим сценаріям користувача. Усі ці етапи адаптовані до вибраного режиму доступності, що гарантує комфортну взаємодію незалежно від індивідуальних особливостей користувача.

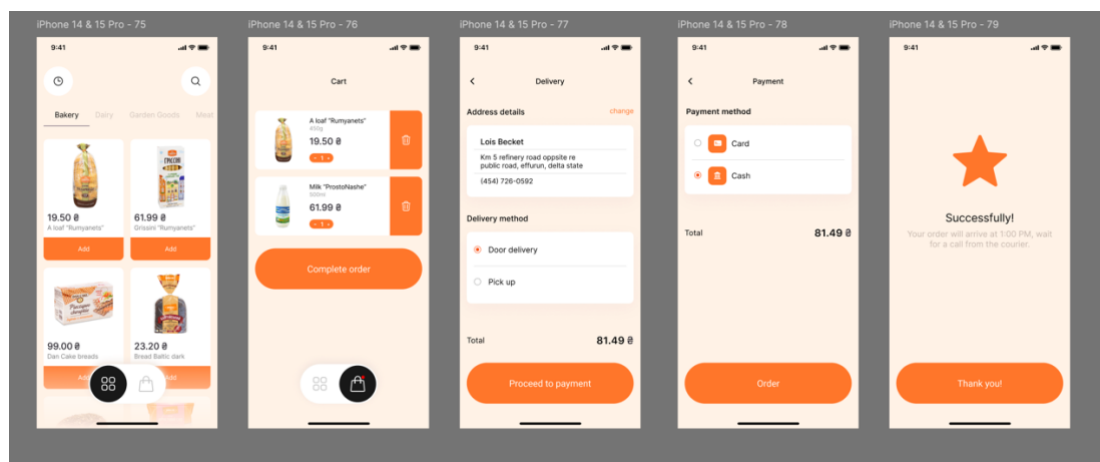


Рисунок 3.15 – Каталог товарів, кошик, оформлення замовлення та зворотній зв’язок

Джерело: розроблено автором на основі виконаного дослідження

Каталог товарів

– Екран каталогу товарів призначений для перегляду асортименту супермаркету, який користувач обирає на попередньому етапі. Візуально товари представлені у форматі плиток, кожна з яких містить:

- зображення продукту;
- його назву;
- ціну;
- кнопку “Add” для додавання у кошик.

У верхній частині екрана реалізовано фільтр категорій (наприклад, хлібний відділ, молочний, м’ясний), що дозволяє користувачеві швидко переключатися між розділами. У прототипі передбачено інтуїтивну навігацію між групами товарів, що зручно для користувачів із когнітивними особливостями. Завдяки адаптивному режиму:

- плитки мають чітку рамку та велику кнопку “Add”, зручно натискати навіть при порушенні моторики;
- у режимі високого контрасту використовуються яскраві акцентні кольори для візуального виокремлення;
- кількість тексту на кожній плитці зведена до мінімуму, що сприяє кращому сприйняттю.

Кошик

Після додавання товарів користувач може перейти до перегляду вмісту кошика. Екран кошика містить:

- список обраних товарів із зазначенням їх кількості;
- ціну за одиницю та загальну суму замовлення;
- кнопку “Complete order” для переходу до оформлення.

Оформлення кошика відповідає принципам спрощеної навігації — усі дії здійснюються на одному екрані без зайвих переходів. У режимі доступності кнопки збільшено, а кольори — оптимізовано для візуального сприйняття.

Оформлення замовлення

Після натискання кнопки “Complete order” система переводить користувача на екран введення даних доставки. Тут користувач вводить:

- адресу;
- тип доставки (кур'єр / самовивіз);
- бажаний час;
- спосіб оплати (готівка або картка).

Цей етап також реалізовано з урахуванням адаптації: поля вводу мають достатній розмір і відступи, назви елементів супроводжуються іконками та кожна дія супроводжується підказкою або коротким описом (для користувачів з когнітивними особливостями).

Після підтвердження замовлення з'являється фінальний екран із повідомленням про успіх і кнопкою “Thank you”, що завершує основний користувацький сценарій.

3.4 Майбутня реалізація

3.4.1 Архітектура майбутньої інформаційної системи (клієнт-серверна модель)

У перспективі повної реалізації інформаційної системи з адаптивним інтерфейсом передбачається використання клієнт-серверної архітектури, яка є найбільш ефективною з огляду на вимоги масштабованості, гнучкості та централізованого управління даними. Така модель дозволить забезпечити зручну взаємодію між користувачем і системою, незалежно від обраного пристрою, типу підключення та режиму доступності.

Функціонування системи базується на чіткій взаємодії між клієнтською частиною, яка відповідає за інтерфейс та взаємодію з користувачем, і серверною частиною, що обробляє запити, здійснює логіку адаптації та забезпечує доступ до бази даних. Користувацький інтерфейс передбачається реалізовувати за допомогою сучасних фреймворків, таких як React або Flutter, що забезпечують кросплатформність та високу швидкість рендерингу. Серверна частина функціонуватиме як REST API або GraphQL-сервіс на базі

Node.js або Python (Django/Flask), що дозволить ефективно обробляти запити користувачів, керувати сесіями, авторизацією та персональними налаштуваннями.

Передбачено, що кожен користувач зможе реєструватися в системі, і під час першого входу вказувати режим доступності, який буде збережений у базі даних. Цей параметр дозволить клієнтській частині автоматично застосовувати відповідні стилі інтерфейсу (збільшення шрифтів, зміна кольорової схеми, спрощення навігації тощо) на основі отриманого профілю. Збереження інформації про користувачів, товари, замовлення та адаптаційні налаштування здійснюватиметься в централізованій базі даних, наприклад, MongoDB або PostgreSQL.

На рівні логіки реалізації можна передбачити, що користувач, обравши адаптивний режим, отримає персоналізований інтерфейс уже при наступному вході. Клієнтська частина при завантаженні застосунку здійснює запит до сервера, отримує відповідь у форматі JSON та, залежно від параметра `accessibility_mode`, вмикає потрібні стилі або сценарії. Наприклад, якщо збережено значення `visual`, система автоматично підвантажує стилі з підвищеним контрастом і великим шрифтом, без потреби в ручному перемиканні режимів.

Навіть попри те, що наразі система реалізована лише у вигляді прототипу в Figma, структура її роботи вже враховує архітектурні особливості майбутнього застосунку. Логіка переходів між екранами в прототипі повністю відповідає очікуваній логіці роутінгу у фронтенді, а структура даних — запланованій моделі взаємодії з сервером. Це дозволяє вже на ранньому етапі протестувати всі функціональні сценарії та забезпечити їх подальшу інтеграцію у реальному програмному середовищі.

Окремі фрагменти умовного коду вже демонструють, яким чином може бути реалізована передача даних між клієнтом і сервером у процесі реєстрації користувача, застосування режиму доступності або отримання каталогу товарів. Наприклад, клієнтська частина, використовуючи метод `fetch`, надсилає дані реєстрації до API, а у відповідь отримує об'єкт користувача з

усіма параметрами профілю. На основі цих даних і здійснюється первинне налаштування інтерфейсу, як наведено на рис. Б1, Б2, Б3, Б4.

Таким чином, запропонована клієнт-серверна модель реалізації інформаційної системи дозволяє забезпечити надійну, масштабовану й адаптивну платформу, яка відповідатиме сучасним вимогам доступності. Поєднання гнучкої архітектури, зручного UI та індивідуалізованого досвіду користувача забезпечує високий потенціал для впровадження системи в практичну діяльність.

3.4.2 Перелік технологій, які можуть бути використані

Для реалізації адаптивної інформаційної системи доцільно обрати сучасний стек технологій, що забезпечить не лише функціональність, а й відповідність вимогам доступності, масштабованість, стабільність і можливість швидкого оновлення інтерфейсу під потреби користувачів з інклюзією.

Таблиця 3.1 – Рекомендовані технології для реалізації інформаційної системи

<i>Технологія / інструмент</i>	<i>Призначення</i>
React	Створення клієнтського інтерфейсу з підтримкою компонентної структури та адаптивної навігації.
React Aria / Headless UI	Забезпечення доступності інтерфейсу згідно зі стандартами WCAG та ARIA.
Node.js + Express	Реалізація серверної логіки, маршрутизації та обробки REST-запитів.
TypeScript	Забезпечення типізації на етапі розробки для підвищення стабільності коду.
Firebase	Хмарна автентифікація, базове зберігання даних, хостинг і швидке розгортання.

Продовження таблиці 3.1

MongoDB / PostgreSQL	Сховище користувацьких даних, товарів, замовлень та адаптаційних параметрів.
Web Speech API	Озвучення елементів інтерфейсу, підтримка голосового супроводу.
CSS Media Queries	Адаптація стилів під різні режими доступності (шрифти, контраст, розміри).
React Native / Flutter	Розробка мобільного застосунку або прогресивного вебзастосунку (PWA).

Джерело: розроблено автором на основі виконаного дослідження

У якості клієнтської частини оптимальним рішенням є використання React або Flutter. React дозволяє створювати компонентну структуру, яка легко підтримується та адаптується під різні сценарії користувача. Крім того, існує велика кількість бібліотек, які підтримують WCAG-стандарти (наприклад, React Aria, Headless UI), що особливо важливо при розробці інклюзивного інтерфейсу.

Для побудови серверної логіки підійде Node.js з фреймворком Express, що дозволяє швидко реалізувати API, необхідні для обробки запитів, реєстрації, авторизації та налаштувань доступності. У якості мови серверного коду можна використати JavaScript або TypeScript — залежно від обраного рівня типізації та вимог до масштабованості.

Для зберігання даних про користувачів, їхні налаштування, список товарів і замовлення можливе використання MongoDB як документно-орієнтованої бази даних, або PostgreSQL — у разі потреби в реляційній структурі. Для малих проєктів або прототипів ефективним рішенням є Firebase, який надає готові модулі для автентифікації, хмарного зберігання та хостингу, а також підтримку реального часу.

Окрему увагу варто приділити технологіям, що забезпечують адаптацію та доступність. Для реалізації голосового супроводу можуть бути використані браузерні API, наприклад, Web Speech API, а для зчитування елементів — ARIA-атрибути. Також у проєкті може бути застосована підтримка керування

з клавіатури, масштабування елементів за допомогою CSS Media Queries та інтеграція з допоміжними технологіями.

Залежно від обсягу функціоналу та цільової платформи, застосунок може бути реалізовано як PWA (Progressive Web App) або мобільний застосунок на базі React Native, що забезпечує доступність з різних пристроїв — як з комп'ютера, так і зі смартфона.

Таким чином, вибір зазначених технологій створює надійну базу для подальшого переходу від прототипу до повноцінної реалізації системи з урахуванням усіх функціональних і нефункціональних вимог, включаючи ті, що стосуються інклюзії та цифрової рівності.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розглянуто, обґрунтовано та реалізовано концепцію інформаційної системи з адаптивним інтерфейсом для користувачів з інклюзією. Актуальність обраної теми зумовлена потребою у створенні цифрових рішень, доступних для людей із різними формами обмежень — порушеннями зору, моторики та когнітивного сприйняття. Враховуючи сучасні міжнародні стандарти доступності (зокрема WCAG 2.1), а також соціальний запит на цифрову інклюзію, розробка такої системи є не лише інженерним викликом, а й суспільно значущим завданням.

У першому розділі було охарактеризовано предметну галузь — цифрові сервіси доставки — та виявлено її потенціал для впровадження адаптивних інтерфейсів. Проведено аналіз наукових джерел і практичних рішень, серед яких — інструменти Apple VoiceOver, Microsoft Accessibility Suite, Google Lookout, а також наукові праці Якоба Нільсена та Дона Нормана. Виявлено, що більшість існуючих рішень не забезпечують динамічної адаптації та не враховують автоматичного профілювання користувача.

У другому розділі сформульовано бізнес-, функціональні та нефункціональні вимоги до інформаційної системи. Проведено постановку задачі, що полягала в побудові адаптивного інтерфейсу, здатного змінюватися під конкретні потреби користувача. Описано вхідні та вихідні дані, зведено функціональні вимоги у структуровану модель та створено відповідну специфікацію.

У межах третього розділу було реалізовано ключовий компонент проєкту — прототип мобільного застосунку DeliveryApp у середовищі Figma. Прототип охоплює повний цикл користувача: від реєстрації та вибору режиму доступності до перегляду товарів, управління кошиком, доставки й оплати. Важливою частиною реалізації стало створення окремих сценаріїв для користувачів з інклюзією: застосовано високий контраст, збільшені елементи керування, спрощену навігацію та мінімізацію текстового навантаження.

Візуально оформлені екрани чітко демонструють адаптацію відповідно до потреб кожної групи користувачів.

Було проаналізовано типи даних, які відображаються в інтерфейсі, способи їх оновлення (імітація запитів), а також можливу архітектуру системи — клієнт-серверна модель із використанням таких технологій, як React, Node.js, Firebase. У підрозділі реалізації було представлено фрагменти коду, які демонструють, як саме може реалізовуватися логіка адаптації інтерфейсу, зокрема реєстрація користувача та застосування адаптивного режиму.

У підсумку, розроблений прототип демонструє, як можна реалізувати базові сценарії користувацької взаємодії з урахуванням індивідуальних потреб людей з інклюзією. Реалізовані адаптивні рішення дозволяють покращити доступність інтерфейсу, зробити навігацію більш зручною, а візуальне оформлення — відповідним до особливостей сприйняття. Цей підхід може бути використаний як основа для подальшого проектування повнофункціональної інформаційної системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Засоби спеціальних можливостей для зору - Підтримка від Microsoft. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/topic/засоби-спеціальних-можливостей-для-зору-b3c57606-e0af-46d2-97b4-fa6b5fba4fa1> (дата звернення: 01.06.2025).
2. Accessibility. Apple. URL: <https://www.apple.com/accessibility/> (дата звернення: 01.06.2025).
3. Accessibility Articles, Videos, Reports, and Training Courses. Nielsen Norman Group. URL: <https://www.nngroup.com/topic/accessibility/> (дата звернення: 06.06.2025).
4. Accessibility. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.0/getting-started/accessibility/> (дата звернення: 06.06.2025).
5. Accessibility Insights. Accessibility Insights. URL: <https://accessibilityinsights.io> (дата звернення: 04.06.2025).
6. Accessibility / MDN. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/Accessibility> (дата звернення: 10.06.2025).
7. Disability. World Health Organization (WHO). URL: <https://www.who.int/news-room/fact-sheets/detail/disability-and-health> (дата звернення: 01.06.2025).
8. Firebase Documentation. Firebase. URL: <https://firebase.google.com/docs> (дата звернення: 07.06.2025).
9. Free Prototyping Tool: Build Interactive Prototype Designs | Figma. Figma. URL: <https://www.figma.com/prototyping/> (дата звернення: 01.06.2025).
10. Home - The A11Y Project. Home - The A11Y Project. URL: <https://www.a11yproject.com> (дата звернення: 03.06.2025).
11. How to meet WCAG (quickref reference). W3C. URL: <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.1&showtechniques=111,126> (дата звернення: 01.06.2025).

12. Inc A. React Aria. React Spectrum Libraries. URL: <https://react-spectrum.adobe.com/react-aria/index.html> (дата звернення: 03.06.2025).
13. Index | Node.js v24.2.0 Documentation. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 04.06.2025).
14. Index | Node.js v24.2.0 Documentation. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 07.06.2025).
15. Innovative behindertengerechte Android-Handys & -Geräte | Android. Android. URL: <https://www.android.com/accessibility/> (дата звернення: 01.06.2025).
16. Material Design. Material Design. URL: <https://m3.material.io> (дата звернення: 05.06.2025).
17. Nielsen J. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (дата звернення: 01.06.2025).
18. Quick Start – React. React. URL: <https://react.dev/learn> (дата звернення: 04.06.2025).
19. WebAIM: Introduction to Web Accessibility. WebAIM: Web Accessibility In Mind. URL: <https://webaim.org/intro/> (дата звернення: 02.06.2025).
20. Web Content Accessibility Guidelines (WCAG) 2.1. W3C. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 01.06.2025).
21. Цифрова інклюзія та доступність: соціальна діджиталізація / За ред. Г. Давиденко. Вінниця: ВНАУ, 2023. URL: https://vsei.vn.ua/images/Doc/Nauka/Inklusivna_osvita/cifrova-inklyuziya-ta-dostupnist-socialna-didzhitalizaciya.pdf (дата звернення: 07.06.2025).
22. Омельчук А. В., Сидоренко В. С. Інклюзивний вебдизайн: стандарти та цифрові інструменти тестування доступності вебсайту. ResearchGate. URL: https://www.researchgate.net/publication/388909816_INKLUZIVNIJ_VEBDIZAJN_STANDARTI_TA_CIFROVI_INSTRUMENTI_TESTUVANNA_DOSTUPNOSTI_VEBSAJTU (дата звернення: 07.06.2025).

23. Davies D. K., Stock S. E., Wehmeyer M. L. Enhancing independent internet access for individuals with intellectual and developmental disabilities. PubMed Central. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7479802/?utm> (дата звернення: 07.06.2025).

24. Stirenko S., Loboda A., Trach O., Lutsenko I., Sachenko A. Adaptive E-learning System User Interfaces for People with Disabilities. CEUR Workshop Proceedings, Vol. 3925, 2024. URL: <https://ceur-ws.org/Vol-3925/paper17.pdf> (дата звернення: 07.06.2025)

ДОДАТКИ

Додаток А

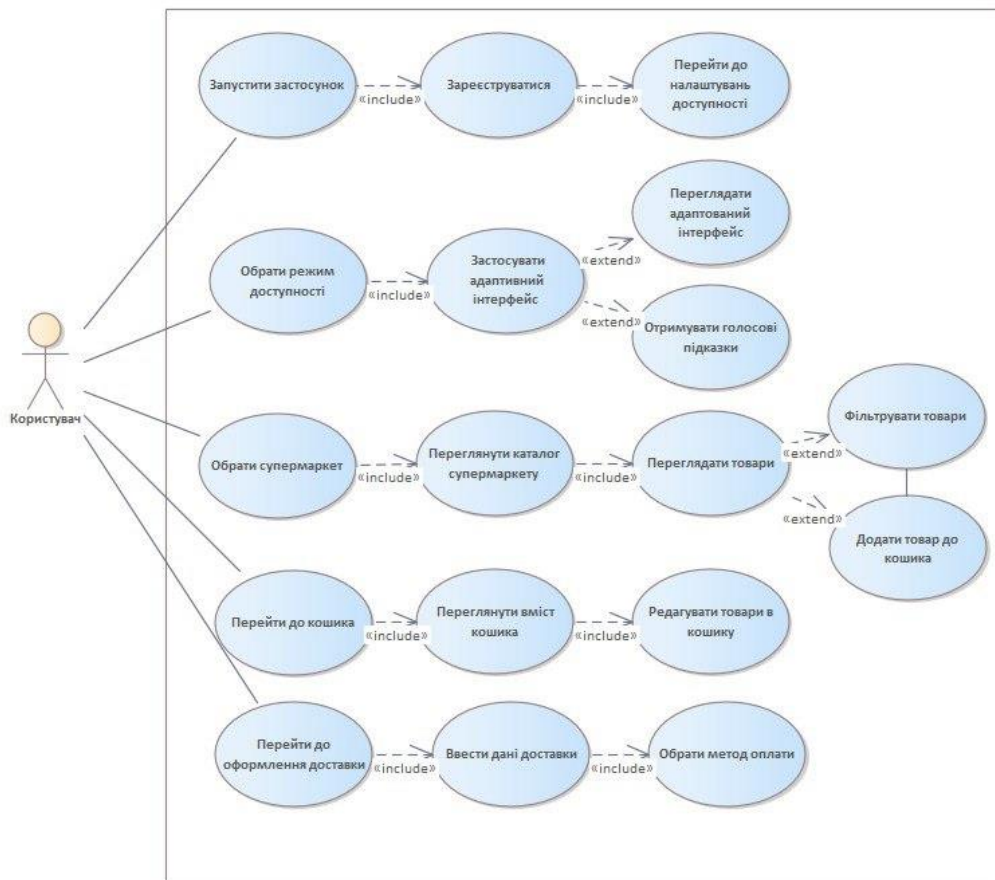


Рисунок А.1 – Діаграма прецедентів

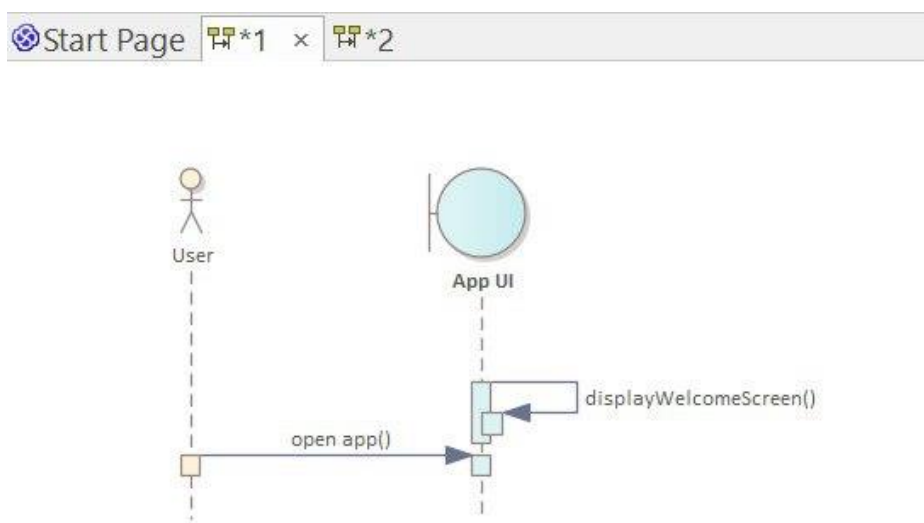


Рисунок А.2 – Діаграма послідовності «Запуск застосунку та виведення привітального екрана»

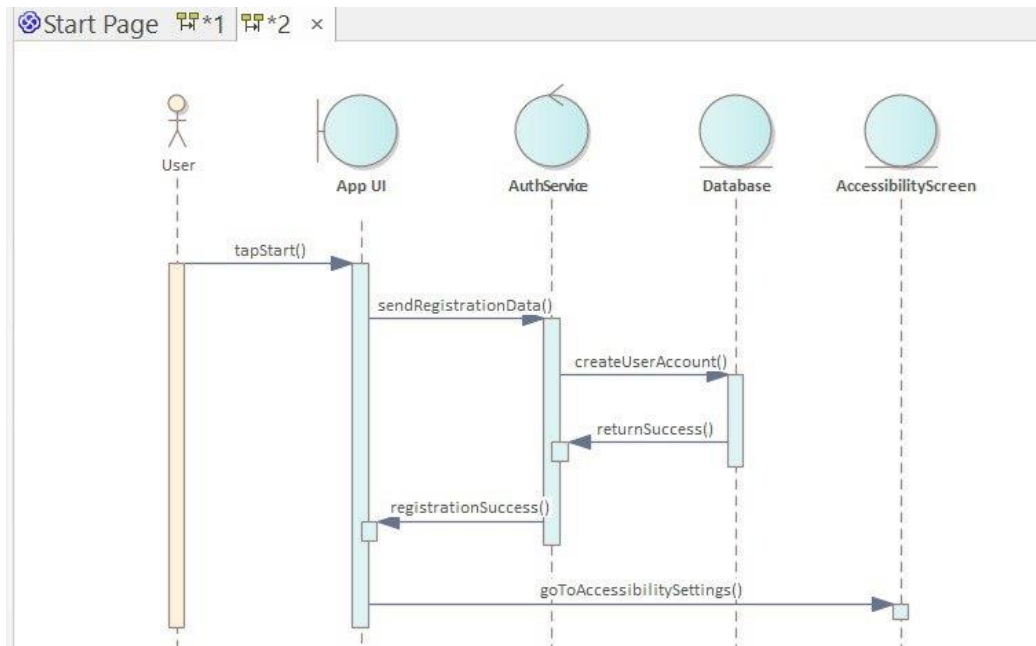


Рисунок А.3 – Діаграма послідовності «Реєстрація нового користувача та перехід до налаштувань доступності»

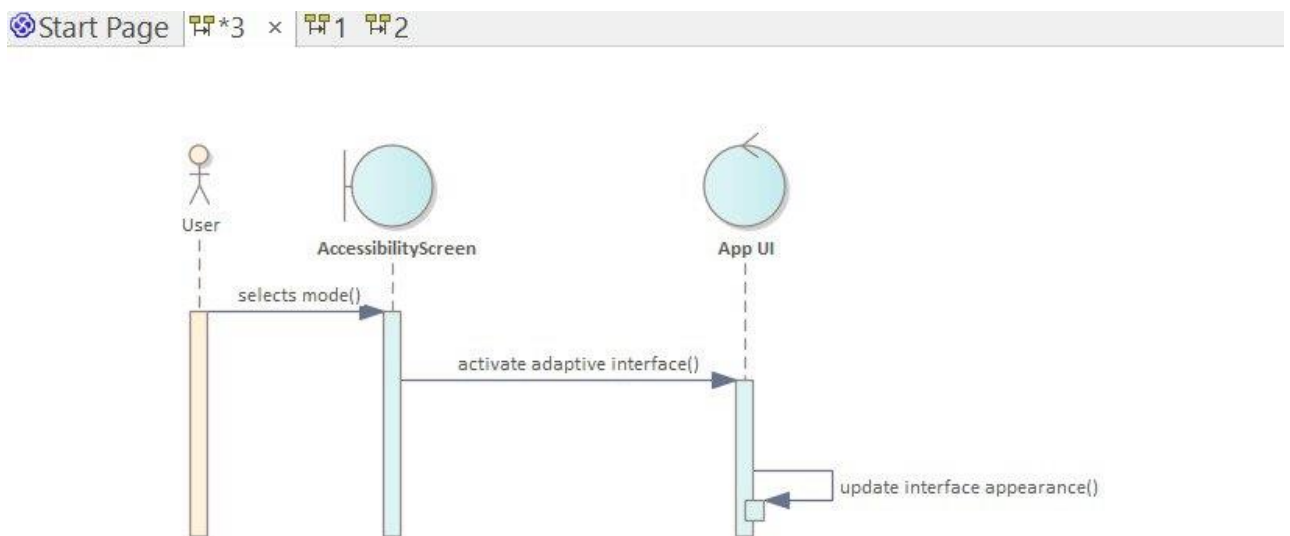


Рисунок А.4 – Діаграма послідовності «Активация адаптивного режиму інтерфейсу»

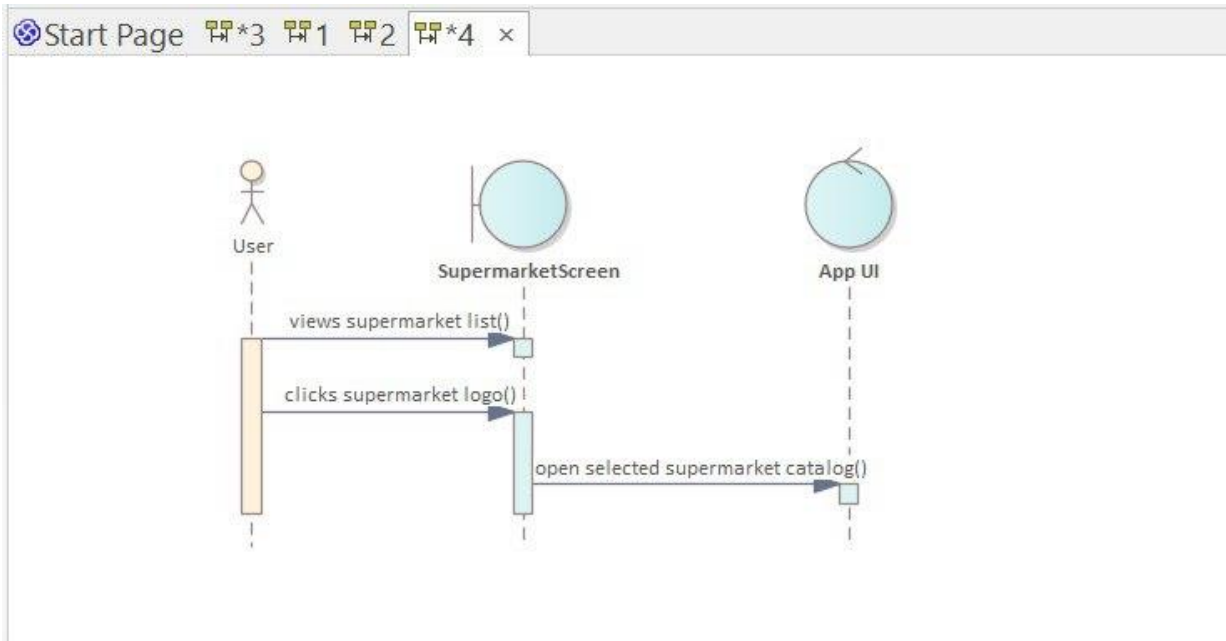


Рисунок А.5 – Діаграма послідовності "Процес відкриття каталогу супермаркету"

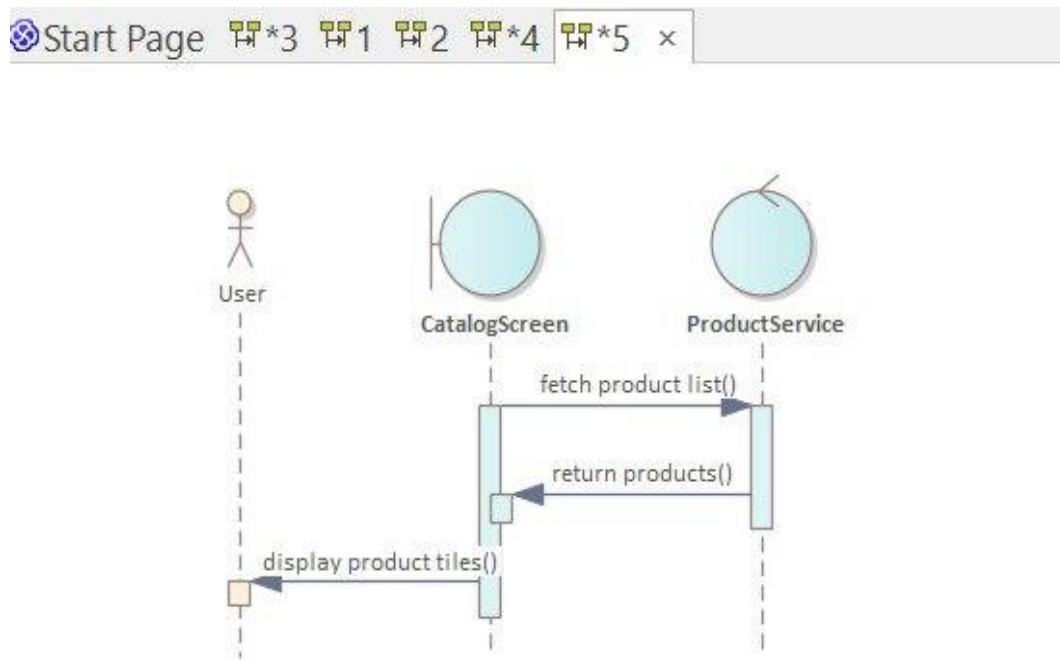


Рисунок А.6 – Діаграма послідовності "Процес відображення списку товарів"

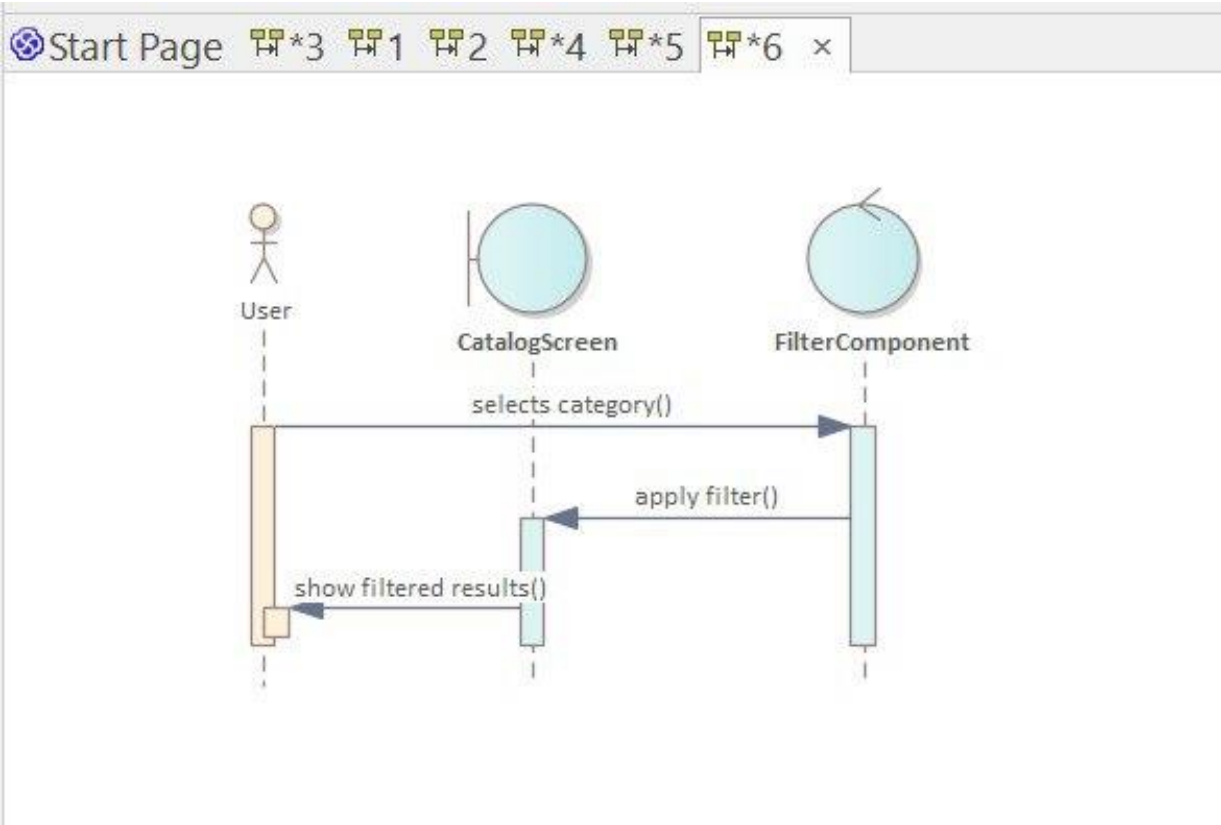


Рисунок А.7 – Діаграма послідовності "Процес фільтрації товарів за категоріями"

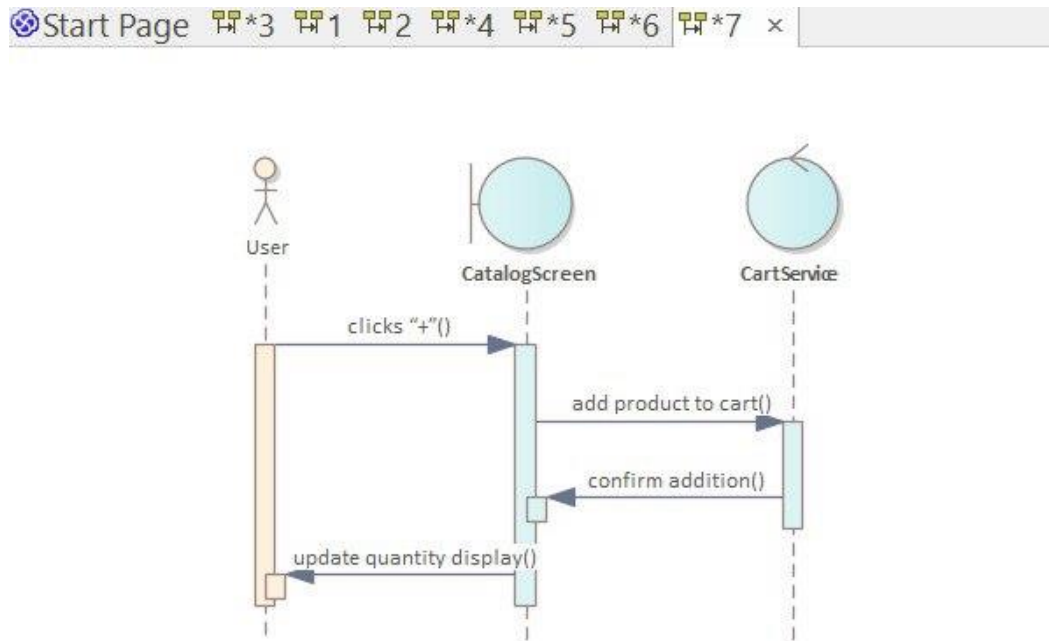


Рисунок А.8 – Діаграма послідовності "Процес додавання товару до кошика"

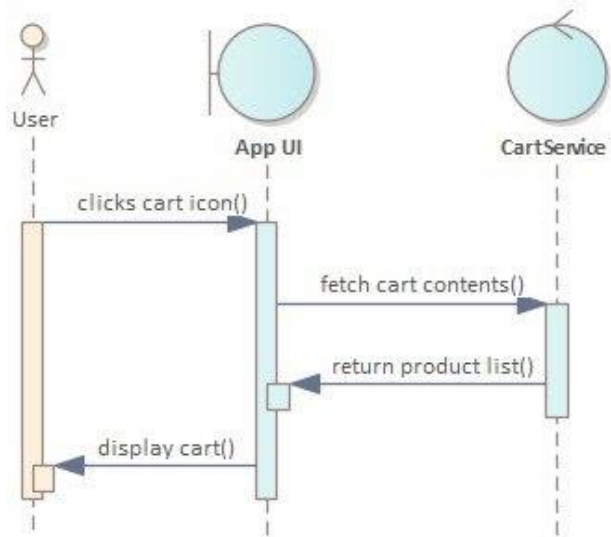


Рисунок А.9 – Діаграма послідовності "Процес відображення вмісту кошика"

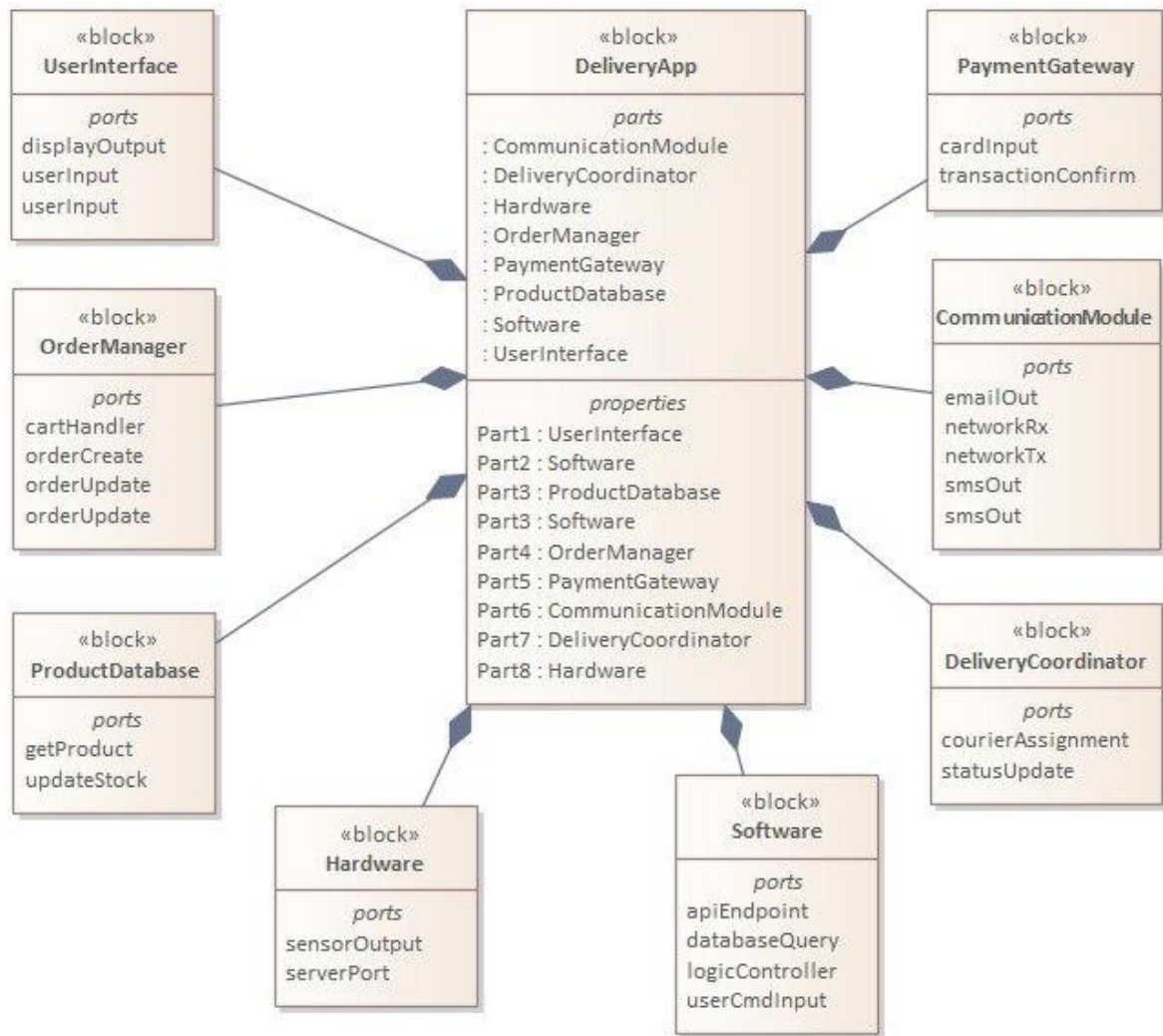


Рисунок А.10 – Діаграма визначення блоків

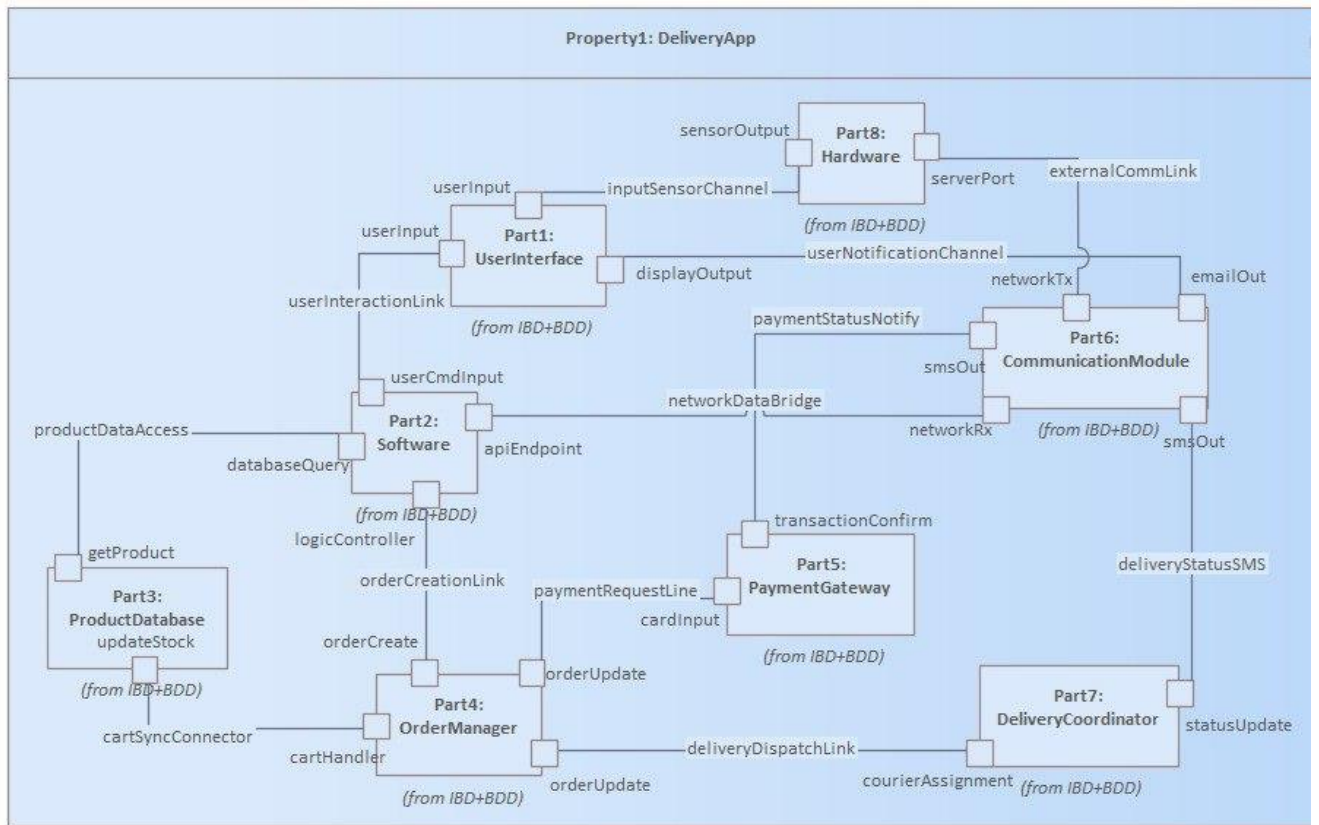


Рисунок А11 – Діаграма внутрішніх блоків

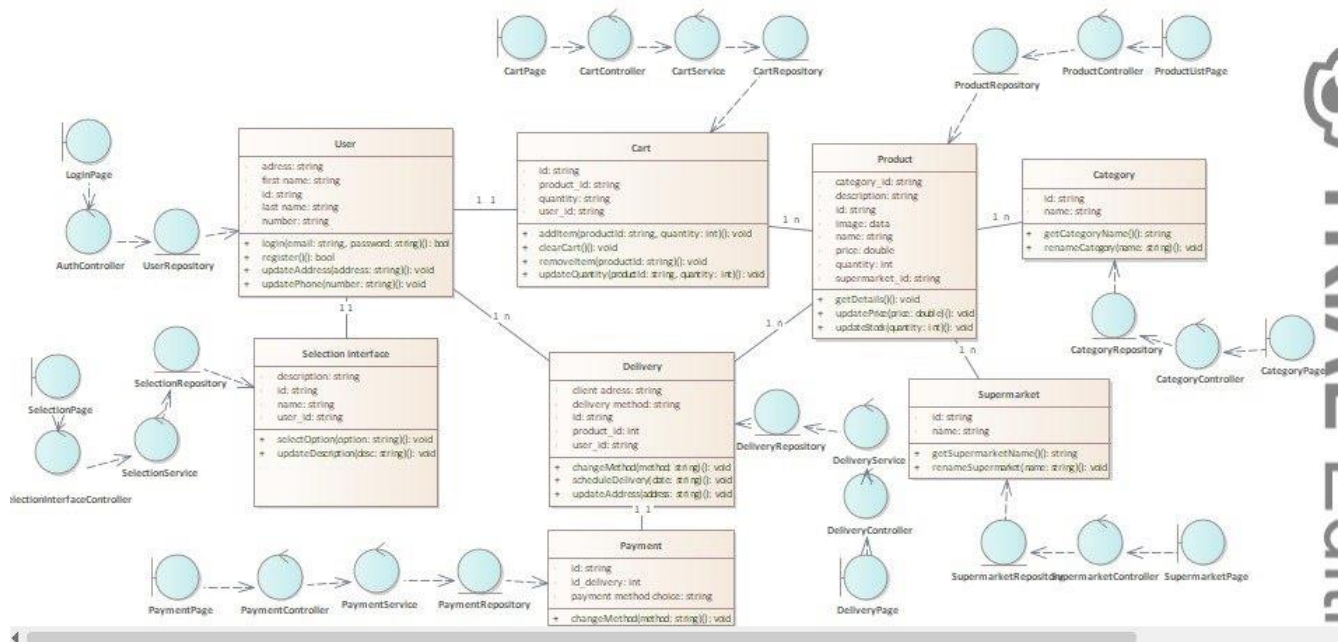


Рисунок А12 – Діаграма класів

Додаток Б

```
const handleRegister = async () => {
  const res = await fetch('https://api.deliveryapp.com/register', {
    method: 'POST',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify({ name, email, password }),
  });

  if (res.ok) {
    navigate('/accessibility');
  }
};
```

Рисунок Б1 – Приклад можливої реалізації клієнтського запиту (React)

```
app.post('/register', async (req, res) => {
  const { name, email, password } = req.body;
  const user = await db.createUser({ name, email, password });
  res.status(201).json(user);
});
```

Рисунок Б2 – Серверна логіка обробки реєстрації (Node.js / Express)

```
{
  "_id": "u83c5e",
  "name": "Anna",
  "email": "anna@gmail.com",
  "password": "hashed_pass",
  "accessibility_mode": "visual"
}
```

Рисунок Б3 – Структура колекції користувачів у базі

```
useEffect(() => {  
  if (user.accessibility_mode === 'visual') {  
    applyVisualMode();  
  } else if (user.accessibility_mode === 'motor') {  
    applyMotorMode();  
  } else if (user.accessibility_mode === 'cognitive') {  
    applyCognitiveMode();  
  }  
}, []);
```

Рисунок Б4 – Динамічне застосування режиму доступності (React)



Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Інформаційна система з адаптивним Інтерфейсом для осіб з інклюзією

Автор

Науковий керівник / Бєсперт

Козачук Дарина РоманівнаРіппа С.П.

Підразділ

кафедра Інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фрази для коефіцієнта подібності 2



КП 2

10631

Кількість слів



КЦ

86058

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		4
Інтервали		0
Мікропробіли		28
Білі знаки		0
Парафрази (SmartMarks)		6

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

Ідентифікаційний номер*	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Кількість центральних слів (ТІМЧ МЕНШЕ)
1	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	38 0.36 %
2	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	20 0.19 %
3	https://ir.kneu.edu.ua/server/api/core/bitstreams/38362e99-7f5f-449c-879b-cb24b773a81d/content	14 0.13 %
4	https://ir.kneu.edu.ua/server/api/core/bitstreams/38362e99-7f5f-449c-879b-cb24b773a81d/content	12 0.11 %
5	https://ir.kneu.edu.ua/server/api/core/bitstreams/38362e99-7f5f-449c-879b-cb24b773a81d/content	10 0.09 %

Рисунок Б.5