Міністерство освіти і науки України
Кам’янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп’ютерних наук

Кваліфікаційна робота бакалавра

з теми:

«РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ КОНТРОЛЮ ТА ПЛАНУВАННЯ ОСОБИСТОГО БЮДЖЕТУ НА СТЕКУ MEAN»

Виконав: здобувач вищої освіти
групи KNms1-B21
спеціальності 122 Комп’ютерні науки
ПРОДОЛЯК Б.О.

Керівник:

СМАЛЬКО О.А., кандидат педагогічних наук, доцент кафедри комп’ютерних наук, доцент

Рецензенти:

СЕМЕНЕЦЬ І.В., декан природничо-економічного факультету, кандидат економічних наук, доцент

СМОРЖЕВСЬКИЙ Ю.Л., завідувач кафедри математики, кандидат педагогічнх наук, доцент

Кам’янець-Подільський – 2024 р.

АНОТАЦІЯ

Продоляк Б.О. Розробка мобільного застосунку для контролю та планування особистого бюджету на стеку MEAN. Кваліфікаційна робота бакалавра на здобуття освітньо-кваліфікаційного рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Кам'янець-Подільський національний університет імені Івана Огієнка, Кам'янець-Подільський, 2024.

У роботі проаналізовано сучасні методи та технологій для розробки мобільних застосунків для управління особистими фінансами. Ретельно досліджено основні потреби користувачів і конкурентні рішення на ринку програмних засобів для фінансового менеджменту. Визначено ключові функціональні та нефункціональні вимоги до розроблюваної системи.

На основі окреслених вимог і сформованої моделі розроблено мобільний застосунок Spike, який забезпечує користувачів зручною та функціональною платформою для управління особистими фінансами, допомагає їм створювати та контролювати власний бюджет, стежити за доходами та витратами, встановлювати фінансові цілі та отримувати аналітичні звіти.

Ключові слова: мобільний застосунок, стек MEAN, MongoDB, Express.js, Angular, Node.js, особистий бюджет, управління фінансами.

ABSTRACT

Prodolyak B.O. Development of a Mobile application for personal budget control and planning using the MEAN Stack. Qualification work of a bachelor for obtaining an educational qualification level of higher education in the specialty 122 "Computer Science". Kamianets-Podilskyi Ivan Ohiienko National University, Kamianets-Podilskyi, 2024.

The article analyzes modern methods and technologies for developing mobile applications for personal finance management. The basic needs of users and competitive solutions in the financial management software market are thoroughly investigated. The key functional and non-functional requirements for the system under development are determined.

Based on the defined requirements and the formed model, the Spike mobile application was developed, which provides users with a convenient and functional platform for managing personal finances, helps them create and control their own budget, monitor income and expenses, set financial goals and receive analytical reports..

Keywords: mobile application, MEAN stack, MongoDB, Express.js, Angular, Node.js, personal budget, financial management.

ЗМІСТ

ВСТУП

У сучасному світі мобільні застосунки стали невід'ємною частиною повсякденного життя, забезпечуючи користувачам доступ до різноманітних сервісів та бажаної інформації у будь-який час і в будь-якому місці. Особливо чутливими для більшості дорослих людей є дані, пов’язані з веденням, плануванням та контролем особистого бюджету. Таку актуальну, ґрунтовну та у зручному форматі інформацію багато-хто хоче мати напохваті на смартфоні. Саме для цього створювався мобільний застосунок у межах виконання даної кваліфікайної роботи бакалавра.

З розвитком технологій, вимоги до функціональності, продуктивності та безпеки мобільних застосунків зростають, що обумовлює необхідність використання сучасних підходів та ефективниз інструментів у їх розробці. Тому вибір технологічного стеку став ключовим аспектом у процесі підготовки до створення застосунку.

Один із популярних стеків для розробки сучасних мобільних застосунків наразі є MEAN стек, який включає MongoDB, Express.js, Angular та Node.js. Цей стек забезпечує інтеграцію між клієнтською та серверною частинами застосунку, що дозволяє досягти високої продуктивності та гнучкості у розробці. Саме його взято за основу розробки.

Об’єкт дослідження: процес розробки мобільних застосунків на стеку MEAN.

Предмет дослідження: методи та засоби розробки мобільних застосунків для управління особистими фінансами на стеку MEAN.

Метою даного дипломного проекту є розробка мобільного застосунку Spike, який забезпечить користувачів зручною та функціональною платформою для управління особистими фінансами.

Для досягнення поставленої мети вирішувалася низка **завдань,** зокрема:

1) дослідити теоретичні основи ведення, планування, контролю особистого бюджету та найбільш ефективні методи формування та використання особистих та фінансових ресурсів;
2) змоделювати систему контролю і планування особистого бюджету, визначити вимоги до створюваного мобільного застосунку та обрати доцільні інструментальні засоби для його розробки;
3) розробити мобільний застосунок для контролю та планування особистого бюджету відповідно до попередньо сформованого технічного завдання.

У процесі виконання даної дипломної роботи використовувалися різноманітні методи дослідження. Теоретичні методи включали аналіз наукової літератури, огляди статей та доповідей з питань управління особистими фінансами та розробки програмного забезпечення. Практичні методи охоплювали розробку архітектури та коду мобільного застосунку Spike з використанням MEAN стеку, тестування функціональних компонентів та забезпечення безпеки даних. Метод порівняльного аналізу дозволив оцінити переваги та недоліки різних технологічних стеків. Експериментальні методи включали тестування працездатності та продуктивності мобільного застосунку в умовах реальної експлуатації, що дозволило виявити та усунути можливі проблеми на ранніх етапах розробки.

У підсумку, даний дипломний проект спрямований на розробку ефективного мобільного застосунку Spike, який відповідатиме сучасним вимогам та стандартам, а також надасть користувачам зручні та корисні інструменти для управління особистими фінансами. Розроблений застосунок матиме потенціал для залучення широкої аудиторії, підвищення фінансової грамотності користувачів та забезпечення їх функціональними можливостями для ефективного контролю власних фінансових потоків.

Структура роботи. Дипломна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і двох додатків.

Апробація реалізованого проєкту відбувалася впродовж наукової конференції студентів і магістрантів Кам’янець-Подільського національного університету імені Івана Огієнка за підсумками науково-дослідної роботи у 2023-2024 навчальному році (9-10 квітня 2024 року), за результатами якої опубліковано тези «Розробка мобільного застосунку для контролю та планування особистого бюджету на стеку MEAN»[24], а також під час попереднього захисту кваліфікаційних робіт, що відбувся 06.06.2024 року.

РОЗДІЛ 1
ОСНОВИ ВЕДЕННЯ, ПЛАНУВАННЯ ТА КОНТРОЛЮ ОСОБИСТОГО БЮДЖЕТУ

1.1 Основи управління фінансами та визначення фінансових цілей

Ведення, планування та контроль особистого бюджету є основними елементами для досягнення фінансової стабільності та довгострокового благополуччя. Цей процес допомагає людям ефективно розподіляти свої доходи та витрати, уникати боргів та досягати фінансових цілей. У цьому розділі розглянемо основні принципи та методи ведення особистого бюджету, підкріплені даними з різних джерел.

Першим кроком до ефективного управління фінансами є усвідомлення власного фінансового стану. Це включає в себе розуміння джерел доходів, основних витрат та залишкових коштів. За даними «Forbes», фінансова свідомість допомагає уникати імпульсивних витрат і дозволяє приймати обґрунтовані фінансові рішення.

Встановлення короткострокових та довгострокових фінансових цілей є важливим елементом планування бюджету. Це можуть бути цілі, такі як створення резервного фонду, накопичення на велику покупку або інвестиції. «Investopedia» рекомендує визначати конкретні, вимірювані, досяжні, актуальні та обмежені у часі цілі (SMART).

Бюджет є інструментом, який допомагає планувати та контролювати витрати. Він включає всі джерела доходів та категорії витрат. Метод 50/30/20, описаний «Bankrate», є одним з популярних підходів до створення бюджету: 50% на необхідні витрати, 30% на особисті витрати та 20% на заощадження [4].

Метод конвертів передбачає розподіл доходів на різні категорії витрат та заощаджень у фізичні або віртуальні конверти. «CNBC» зазначає, що цей

підхід допомагає краще контролювати витрати, оскільки дозволяє бачити, скільки грошей залишилося в кожному конверті [6].

Один з найпростіших способів забезпечити регулярні заощадження – автоматизувати процес відкладання грошей. Багато банків та фінансових застосунків дозволяють налаштувати автоматичні перекази на заощаджувальні рахунки. «NerdWallet» рекомендує встановлювати автоматичні перекази на заощадження одразу після отримання доходу.

Сучасні технології значно спрощують процес ведення бюджету. Такі застосунки, як Mint, YNAB та Personal Capital, пропонують інструменти для відстеження витрат, планування бюджету та аналізу фінансового стану. «Harvard Business Review» підкреслює, що користувачі застосунків для ведення бюджету мають більшу ймовірність досягти фінансових цілей завдяки постійному моніторингу витрат [8].

Регулярний аналіз витрат допомагає виявити непотрібні або завищені витрати. Це може бути зроблено вручну або за допомогою застосунків. «The Balance» рекомендує щомісячно переглядати витрати та коригувати бюджет, щоб уникати непередбачених витрат.

Після аналізу витрат важливо знайти способи їх оптимізації. Це може включати зменшення витрат на непотрібні послуги, пошук альтернативних дешевших варіантів або перегляд підписок. «Financial Times» радить порівнювати ціни на товари та послуги, щоб зекономити.

Управління боргами є критично важливим для фінансової стабільності. Необхідно розробити план погашення боргів, починаючи з тих, що мають найвищі відсоткові ставки. Dave Ramsey, відомий фінансовий експерт, пропонує метод "сніжного кома" для швидшого погашення боргів, який полягає в поступовому збільшенні платежів по найменших боргах.

Сучасні технології надають безліч можливостей для більш ефективного управління особистими фінансами. Мобільні застосунки для ведення бюджету, такі як Mint, YNAB, Personal Capital та інші, пропонують користувачам зручні інтерфейси для ведення бюджету [10], відстеження

витрат та планування фінансів. Вони часто мають функції автоматичного імпорту транзакцій з банківських рахунків та кредитних карт, що спрощує процес моніторингу витрат.

Використання онлайн-банкінгу дозволяє легко контролювати свої фінанси, здійснювати платежі та перекази, а також отримувати сповіщення про баланс та транзакції. Багато банків також пропонують функції бюджетування та аналізу витрат.

Інвестиційні платформи, такі як Robinhood, E*TRADE, Betterment та інші, надають можливість легко інвестувати в акції, облігації та інші фінансові інструменти [7]. Вони також пропонують інструменти для аналізу інвестицій та управління портфелем.

Використання хмарних сервісів для зберігання фінансових даних забезпечує їхню доступність з будь-якого пристрою. Це дозволяє легко синхронізувати дані між різними застосунками та пристроями, забезпечуючи актуальність та безпеку інформації.

Ведення, планування та контроль особистого бюджету є невід'ємною частиною досягнення фінансової стабільності та благополуччя. Використання структурованих методів та сучасних технологій дозволяє ефективніше управляти фінансами, досягати поставлених цілей та забезпечувати фінансову безпеку. Регулярний аналіз та оптимізація витрат, автоматизація заощаджень та використання фінансових рішень є ключовими складовими успішного фінансового планування.

1.2 Методи формування та використання особистих фінансових ресурсів

Дослідження ефективних методів формування та використання особистих та фінансових ресурсів є невід'ємною частиною кожної людини. Це шлях до фінансової стабільності, добробуту та особистого зростання. У цьому дослідженні ми поглибимося в різноманітні стратегії та методи, що допомагають керувати особистими та фінансовими ресурсами.

Створення бюджету – перший аспект до успішного фінансового планування. Це складний процес, що вимагає уваги до деталей. Він включає аналіз поточних доходів та витрат, встановлення фінансових цілей та розподіл грошових потоків на різні категорії. Далі йде планування витрат , це важливий етап у керуванні особистими фінансами. Воно допомагає раціоналізувати витрати, підтримуючи баланс між потребами та бажаннями. Основним завданням цього процесу є встановлення пріоритетів та раціональне використання доступних ресурсів.

Освіта та саморозвиток – ключові складові успішного фінансового управління. Чим більше ми знаємо про фінансові інструменти та стратегії, тим краще ми можемо керувати своїми фінансами. Самоосвіта, участь у фінансових семінарах та читання відповідних книг допомагають покращити фінансову грамотність та приймати обдумані рішення.Також одним із важливих методів є ефективне використання фінансових інструментів, який полягає в оптимізації особистих ресурсів. Інвестування, заощадження та управління боргами допомагають забезпечити фінансову стабільність та максимізувати доходи в майбутньому.

Створення бюджету вимагає таких пунктів:

- Визначення доходів: Включає всі джерела доходу, такі як зарплата, додатковий заробіток, дивіденди та інші надходження.
- Оцінка витрат: Включає регулярні витрати (оренда, комунальні послуги, транспорт), змінні витрати (харчування, розваги) та рідкісні витрати (відпустка, великі покупки).
- Розподіл витрат на категорії: Дозволяє чітко бачити, куди йдуть гроші, і визначити області для оптимізації.
- Встановлення фінансових цілей: Включає короткострокові (купівля техніки), середньострокові (накопичення на відпустку) та довгострокові (накопичення на пенсію) цілі.
- Регулярний перегляд бюджету: Дозволяє коригувати бюджет відповідно до змін у доходах та витратах.

Другим важливим аспектом є заощадження. Це допомагає забезпечити фінансову безпеку та накопичення ресурсів для майбутніх потреб. Заощадження можна здійснити шляхом встановлення мети заощадження, визначення стратегії заощадження та автоматизації процесу.Є різні методи заощадження, але найефективнішими є до прикладу метод 50/30/20, який полягає в тому, що 50% доходу йде на необхідні витрати, 30% – на бажані витрати, і 20% – на заощадження та інвестиції; метод автоматизації зощаджень є дійсно зручним способом налаштування правильного та ефективного бюджету,адже його суть полягає у налаштуванні автоматичного перерахування частини доходу на заощаджувальний рахунок

Третім аспектом є інвестування. Інвестування дозволяє не лише зберігати, але й примножувати кошти. Існують різні види інвестицій, такі як акції, облігації, нерухомість та інші. Вибір конкретного виду інвестицій залежить від індивідуальних фінансових цілей та ризикових уподобань.

Для ефективного управління фінансами також важливо вміти керувати боргами. Управління боргами передбачає аналіз поточних заборгованостей, розробку плану погашення боргів та уникнення нових зобов'язань.

Нарешті, використання сучасних технологій може значно спростити фінансове планування та управління ресурсами. Мобільні застосунки для управління фінансами, онлайн-банкінг та інші інструменти дозволяють легко відстежувати витрати, створювати бюджети та встановлювати фінансові цілі.

Використання особистих та фінансових ресурсів є ключовим аспектом у житті кожної людини, оскільки воно дозволяє максимізувати їх потенціал та забезпечує ефективне використання. Одні з головних методів використання фінансових ресурсів наведенні нижче:

– Планування є надзвичайно важливим методом використання ресурсів. Це включає в себе створення розкладу та встановлення конкретних цілей, що допомагає організувати час та гроші для досягнення намічених результатів.

– Бюджетування допомагає контролювати фінанси, розподіляючи доходи на різні категорії витрат. Це дозволяє уникнути перевитрат та забезпечує фінансову стабільність.

– Ефективне використання доходів включає розумний розподіл грошей на основні потреби, заощадження та інвестування для забезпечення майбутніх потреб.

– Заощадження допомагає створити фінансовий резерв для непередбачених витрат та забезпечує фінансову безпеку.

– Інвестування є методом для примноження грошей, що може включати вкладення коштів у різні фінансові інструменти.

– Управління боргами передбачає аналіз та стратегічне погашення боргів, уникання надмірних зобов'язань та вчасне погашення.

– Розвиток фінансової грамотності допомагає краще розуміти принципи управління ресурсами та приймати обдумані фінансові рішення.

Ці методи сприяють ефективному використанню особистих та фінансових ресурсів і досягненню фінансової стабільності та добробуту.

1.3 Підходи до планування особистого бюджету та витрат

Планування особистого бюджету та витрат є найважливішим аспектом у досягненні фінансової стабільності та забезпеченні ефективних витрат. Вони охоплюють кілька перевірених способів, які допомагають людям ефективно керувати своїми фінансами. Ці методи забезпечують структурований підхід до розподілу доходів та контролю витрат, сприяючи фінансовій стабільності та досягненню фінансових цілей.

Правило 50/30/20

Одним з найпопулярніших методів бюджетування є правило 50/30/20, розроблене сенатором Елізабет Воррен та її дочкою Амелією Воррен Тайагі у книзі "All Your Worth: The Ultimate Lifetime Money Plan". Це правило передбачає розподіл доходів на три основні категорії: 50% доходів

спрямовуються на основні потреби, такі як житло, харчування, транспорт та комунальні послуги; 30% на особисті витрати, включаючи розваги, хобі та покупки для себе; і 20% на заощадження та погашення боргів.

Згідно з дослідженням, проведеним Гарвардською школою бізнесу, таке розподілення допомагає підтримувати баланс між поточними витратами та майбутніми фінансовими цілями, зменшуючи ризик перевитрат та накопичення боргів. Впровадження цього методу дозволяє користувачам уникати фінансового стресу і забезпечує структурований підхід до управління доходами.

Правило 6 глечиків (JARS budgeting)

Правило 6 глечиків або JARS budgeting передбачає розподіл доходів на шість різних "глечиків" або категорій: 55% доходів на основні потреби, 10% на довгострокові заощадження, 10% на розваги, 10% на освіту та розвиток, 10% на інвестиції та 5% на благодійність. Цей метод, популяризований Т. Харвом Екером у його книзі "Secrets of the Millionaire Mind" [12], надає більш деталізований підхід до фінансового планування, допомагаючи розподілити кошти на різні важливі аспекти життя.

Дослідження показують, що такий підхід сприяє не тільки задоволенню поточних потреб, але й розвитку особистості та досягненню довгострокових фінансових цілей. Він допомагає користувачам створювати збалансовані фінансові плани, що враховують як короткострокові, так і довгострокові потреби.

Метод 60-10-10-10-10 (Microsoft budgeting)

Метод 60-10-10-10-10, розроблений компанією «Microsoft», передбачає розподіл доходів таким чином: 60% на основні витрати, 10% на заощадження, 10% на інвестиції, 10% на особисті витрати та 10% на благодійність. Цей метод допомагає забезпечити значний обсяг коштів для основних потреб, зберігаючи при цьому баланс між заощадженнями, інвестиціями та благодійністю.

Згідно з дослідженням «Microsoft Personal Finance», цей підхід дозволяє уникнути перевитрат і сприяє фінансовій стабільності в довгостроковій перспективі. Він є простим і ефективним способом управління особистими фінансами, який може бути адаптований до різних рівнів доходів.

Метод 5 конвертів

Метод 5 конвертів є ще одним популярним підходом до бюджетування, який передбачає фізичне або віртуальне розподілення доходів на п'ять конвертів: для основних витрат, заощаджень, розваг, непередбачених витрат та благодійності. Кожен конверт містить певну суму грошей, яку можна витратити тільки на відповідну категорію.

Дослідження, опубліковане в «Journal of Financial Counseling» and Planning [14], показує, що цей метод допомагає контролювати витрати, зменшуючи ризик перевитрат у конкретних категоріях. Метод 5 конвертів сприяє кращому розумінню фінансових пріоритетів та дозволяє ефективно планувати бюджет.

Кожен з цих методів має свої переваги і може бути адаптований відповідно до індивідуальних фінансових потреб та цілей. Використання структурованого підходу до планування особистого бюджету допомагає досягти фінансової стабільності, зменшити стрес, пов'язаний з фінансами, та забезпечити досягнення довгострокових фінансових цілей. За даними дослідження «National Bureau of Economic Research», систематичний підхід до фінансового планування значно підвищує рівень фінансової грамотності та здатність досягати фінансових цілей.

Розглянуті методи планування особистого бюджету та витрат є основою для досягнення фінансової стабільності та забезпечення ефективного управління фінансами. Однак, ці методи можуть бути адаптовані та розширені для управління більш комплексними бюджетами, такими як сімейний або командний бюджет.

Планування сімейного бюджету включає в себе узгодження фінансових цілей та витрат всіх членів родини. Це забезпечує більш ефективне використання спільних ресурсів та допомагає уникнути конфліктів, пов'язаних з фінансовими питаннями. Наприклад, правило 50/30/20 може бути застосоване до сімейного бюджету з врахуванням доходів та витрат кожного члена сім'ї. Це допомагає забезпечити покриття основних потреб сім'ї, задоволення особистих потреб кожного члена родини та накопичення заощаджень для спільних цілей.

Дослідження, опубліковане в «Journal of Marriage and Family», показує, що спільне планування бюджету позитивно впливає на фінансову стабільність сім'ї та зміцнює сімейні відносини. Спільні фінансові цілі сприяють більшому розумінню та співпраці між членами родини.

Командний бюджет, або бюджет команди, важливий для спільних проєктів або підприємств, де необхідно координувати фінансові ресурси кількох осіб або підрозділів. Тут застосування методів бюджетування допомагає визначити пріоритети витрат, планувати ресурси та забезпечувати фінансову прозорість. Метод 60-10-10-10-10, наприклад, може бути адаптований для командного бюджету, де 60% ресурсів йдуть на основні операційні витрати, 10% на розвиток та інновації, 10% на інвестиції у проєкт, 10% на винагороди членам команди та 10% на соціальні ініціативи.

Дослідження, проведене «Harvard Business Review», підкреслює, що ефективне фінансове планування та управління ресурсами в команді значно підвищує продуктивність та успішність проектів. Планування бюджету на командному рівні забезпечує контроль за витратами, стимулює інновації та сприяє досягненню спільних цілей.

Узагальнюючи, методи планування особистого бюджету є універсальними інструментами, які можуть бути адаптовані для управління як сімейними, так і командними бюджетами. Вони забезпечують структурований підхід до розподілу ресурсів, сприяють фінансовій стабільності та допомагають досягати фінансових цілей. Подальший

розвиток проєкту може включати створення інструментів та застосунків, що підтримують планування та управління фінансами на рівні родини чи команди, інтегруючи сучасні технології для покращення фінансової прозорості та ефективності.

Згідно з аналізом, проведеним «Pew Research Center», інтеграція цифрових рішень у фінансове планування значно підвищує ефективність управління фінансами та дозволяє краще контролювати витрати. Використання таких методів може сприяти не лише особистій фінансовій стабільності, але й зміцненню фінансового здоров'я сім'ї та команд.

РОЗДІЛ 2
ТЕХНОЛОГІЇ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

2.1 Аналіз можливостей існуючих рішень

Мобільна розробка – це робота, пов’язана з розробкою застосунків для мобільних пристроїв. Перелік завдань, до яких часто відноситься мобільна розробка, може включати також мобільний інжиніринг, розробку мобільного контенту, взаємодію з клієнтами, конфігурація безпеки мережі тощо.

Розробники інтерфейсу мають справу з макетом і візуальними зображеннями мобільного застосунку, тоді як розробники заднього виду вирішують проблеми функціональності застосунку.

Існує багато інструментів із відкритим кодом для розробки мобільних рішень, таких як BerkeleyDB, GlassFish, LAMP (Linux, Apache, MySQL, PHP) і Perl / Plack. Це дозволило мінімально знизити вартість навчання мобільної-розробки.

Застосунки для контролю за бюджетом та фінансами та їх планування є важливим елементом життя багатьох людей, які ведуть особистий фінансовий облік. Це робиться не просто так, а для того, щоб керувати власними грошима та досягти певних цілей. Так само, як компаніям потрібен облік для досягнення прибутковості, родинам і людям важливо вести фінансовий облік для контролю та реалізації своїх планів [17].

Ведення фінансового обліку не є складним завданням. Це, як і будь-яка звичка, вимагає внутрішньої мотивації та бажання покращити своє фінансове становище. Спочатку це може здаватися складним, але вже через місяць ви витрачаєте лише кілька хвилин на день для обліку власних фінансів. Пізніше, ви можете використовувати зібрану статистику, щоб змінити некорисні фінансові звички або збільшити свій дохід.

Персональні фінанси не дадуть відповідь на запитання «де взяти гроші?», але вони можуть допомогти з кращим розумінням звідки у вас

появляються кошти й куди вони витрачаються, а вже на основі цієї інформації кожен будує власні висновки.

Корисність ведення фінансів – це перш за все розуміння того як і за що ми живемо, правильні висновки від ведення фінансів – жити відповідно до власних можливостей. Якщо людина хоче жити (витрачати більше) краще або ж заощаджувати більше – вона повинна заробляти більше або ж бюджетувати на перспективу з контролем «некорисних» витрат.

Підходи різні і кожен виділяє свою користь від ведення персональних фінансів. Деякі з плюсів ведення обліку власних доходів й витрат, які можуть слугувати для родини мотивацією: Коли людина почне вести фінансовий облік, її фінансовий стан може стабілізуватись або й покращитись (залежно від «запущеності» поточного стану її коштів). Людина дізнається багато нового про рух власних коштів. Вона дізнається скільки грошей витрачає протягом місяця. Також дізнається, що насправді витрачає не таку суму коштів як їй здається (а швидше за все більше, ніж вона думає). У неї перед очима буде наглядна картина того, куди йдуть її гроші. Вона дізнається, які статті витрат з'їдають більшу частину грошей, а які – меншу. Вона почне тримати свої фінанси під контролем. Дуже багато хто цього не робить і тому не знає, куди зникають їхні гроші. Дивлячись на свій фінансовий звіт, людина почне замислюватися про цінність грошей: чи варто зайвий раз витрачати гроші на чергову непотрібну імпульсивну покупку. Вона буде в курсі свого фінансового стану і буде знати як з часом він змінюється.

На сьогоднішній день створена велика кількість застосунеів подібного плану, але вони мають певні недоліки. Проаналізувавши більшість з них можна виокремити декілька недоліків, а саме:

- *Безпека даних*: Деякі застосунки можуть мати проблеми з безпекою даних, особливо якщо вони підключаються до банківських акаунтів або зберігають чутливу інформацію. Це може стати приводом для зловживання даними користувачів або навіть втрати фінансових активів.

- *Неодноразове оновлення*: Деякі застосунки можуть потребувати частого оновлення, що може бути розчаровуючим для користувачів. Це може призвести до перерв у роботі або вимагати великої кількості часу на оновлення програмного забезпечення.
- *Обмежена функціональність*: Окремі застосунки можуть мати обмежену функціональність або не надавати користувачам достатньо можливостей для управління їх фінансами. Це може призвести до необхідності використання додаткових інструментів або пошуку альтернативних рішень.
- *Несправність інтеграції*: Деякі застосунки можуть мати проблеми з інтеграцією з іншими фінансовими програмами або банківськими системами, що може ускладнити використання застосунку та вимагати додаткових зусиль з боку користувача для вирішення проблем.
- *Високі вимоги до обладнання*: Певні застосунки можуть вимагати великої кількості ресурсів комп'ютера або мобільного пристрою, що може призвести до повільної роботи або перегріву пристрою. Це може стати проблемою для користувачів зі старими або застарілими пристроями.
- *Застарілий дизайн*.

До таких застосунків належать:

Mint – дозволяє користувачам об'єднувати всі свої банківські рахунки, кредитні картки, рахунки за комунальні послуги та інвестиції в одному місці. Програма автоматично класифікує транзакції, пропонує бюджетні рекомендації, нагадує про майбутні рахунки та надає безкоштовний доступ до кредитного рейтингу. Але головні недоліки Mint полягають в тому що в ньому є проблеми з синхронізацією рахунків, відсутність гнучкості у налаштуванні категорій витрат, рекламні оголошення.

За даними дослідження веб-ресурсу «Mint.com: More than just budgeting», Mint є одним із найбільш завантажуваних фінансових застосунків у США з більш ніж 25 мільйонами користувачів.

YNAB (You Need a Budget) – орієнтований на принципи "Zero-Based Budgeting", де кожен долар має своє призначення. Застосунок дозволяє користувачам вручну вводити транзакції, синхронізувати рахунки, створювати бюджети для різних категорій та надавати докладну звітність про фінансове становище. Недоліки цього мобільного застосунку полягають у високий ціні підписки, складність початкового налаштування, відсутність автоматичного класифікування транзакцій. Згідно з веб-ресурсу «You Need a Budget: A Game-Changer in Personal Finance», YNAB має велику кількість позитивних відгуків та користується популярністю серед користувачів, що прагнуть покращити свої фінансові навички.

Підсумовуючи, за даними «Statista», кількість користувачів мобільних фінансових застосунків зросла з 30 мільйонів у 2015 році до 80 мільйонів у 2020 році. Це свідчить про зростаючий інтерес та необхідність у таких інструментах серед населення. Згідно з дослідженням «Grand View Research», ринок фінансових технологій, включаючи мобільні фінансові застосунки, прогнозується зрости до 460 мільярдів доларів до 2025 року.

2.2 Важливі аспекти розробки мобільних застосунків

Порівняння і протиставлення мобільних пристроїв і їхніх застосунків з їх настільними і серверними аналогами є важливою темою і саме завдяки цьому створюються дійсно якісні застосунки. Тому зараз буде доречно перерахувати те, що робить мобільний застосунок по справжньому якісним.

Час запуску

Важливою характеристикою мобільних застосунків є час їх запуску. Оскільки мобільними пристроями зазвичай користуються часто і протягом нетривалих проміжків часу, здатність мобільного застосунку до швидкого запуску є обов'язковою вимогою. Можливо, 6-секундне споглядання заставки текстового процесора, енциклопедії або середовища розробки в очікуванні появи основного вікна відповідної програми і приносить мало задоволення,

однак, з урахуванням загальної тривалості типового сеансу роботи з такими застосунками, цей фактор можна вважати малозначним.

У разі ж мобільного застосування, яке користувач збирається використовувати протягом 20 секунд для уточнення або поновлення невеликої порції інформації, втрата 6 секунд представляється недозволеною розкішшю. Розумно керуватися мнемонічним правилом, згідно з яким тривалість сеансу роботи користувача із застосунком повинна набагато перевершувати тривалість запуску цього застосунка. У разі програмних рішень для настільних комп'ютерів робочі сеанси тривають довше, і тому користувачі готові миритися з періодами очікування більшої тривалості. У разі ж мобільних застосунків тривалість робочих сеансів менше, і тому висувається вимога, щоб тривалість періоду запуску програми була відповідно меншою. Тривалість періоду запуску застосунку є важливим чинником і у випадку настільних комп'ютерів, але у випадку мобільних пристроїв, які характеризуються нерегулярністю і короткочасністю робочих сеансів, цей чинник набуває вирішальне значення.

Відгук пристрою

Сприймаючи мобільні пристрої як слухняні механічні іграшки, люди очікують від них відповідної поведінки. Коли користувач впливає на маніпулятор, натискає на кнопку або будь-яким іншим чином маніпулює елементами керування пристрою, він розраховує на фізичну реакцію з його боку. Не отримавши негайної реакції з боку пристрою, нетерплячий користувач починає нервувати і тут же повторює спробу виконання потрібної операції.

Якщо повторна спроба маніпуляції елементом управління, виконання клацання чи іншого аналогічного дії будуть оброблені вашим застосунком або, помилково, іншим, то в результаті цього можуть виникнути проблеми. Тому вкрай важливо дбати про те, щоб після виконання яких-небудь дій з пристроєм користувач отримував від нього негайне підтвердження реакції на ці дії в тій чи іншій формі.

Ідеальним варіантом такого підтвердження є виконання запитаної операції; краще цього нічого бути не може. На другому місці в цьому відношенні стоїть підтвердження того, що запит отримано і обробляється у фоновому режимі, в той час як застосунок зберігає здатність сприймати інші запити. Третє місце належить відображенню курсора очікування, який подаватиметься користувача про те , що запит обробляється; застосунок на жодні подальші запити не реагує, але користувач знає, що роботу з обслуговування його запиту розпочато і вона виконується. Найгірший з варіантів – це коли користувач не бачить ніякої відповідної реакції з боку пристрою і йому залишається лише здогадуватися про те, сприйнятий його запит чи ні. Як і у випадку інших характеристик, які ми будемо вивчати, ці вимоги не є унікальними для мобільних пристроїв, але в даному випадку вони набувають особливого значення в силу специфіки роботи з цими пристроями і того, чого люди від них чекають.

Однаковість стилю інтерфейсу

Враховуючи те що будь-який мобільний пристрій відрізняється компактністю і самодостатністю, бажання користувачів працювати з інтерфейсом, який не залежить від характеру розв'язуваних за допомогою пристрою завдань, видаються цілком природними. Кожний мобільний пристрій має власний функціональний колорит. Типовий вдало спроєктований застосунок сприймається не як розрізнений набір засобів, а як гармонійне розширення можливостей самого пристрою.

З цієї причини при проєктуванні застосунків для мобільних пристроїв дуже важливо дотримуватися єдиного стилю. Способи запуску і зупинки застосунків, переміщення в межах інтерфейсу, а також організації діалогів між користувачем і застосунками повинні ретельно продумувати окремо для кожного типу пристроїв. Користувачі мобільних пристроїв підсвідомо підлаштовуються під метафори користувача інтерфейсу, і будь-які відхилення від звичних шаблонів доставляють їм відчутні незручності. Створення звичної робочого середовища багато значить і в разі застосунків для

настільних комп'ютерів, але в силу різноманітності можливостей, які пропонуються такими застосунками, вони дозволяють добитися однієї і тієї ж мети декількома способами (наприклад, за допомогою сполучень клавіш, клацань мишею, меню і панелей інструментів). У разі ж мобільних пристроїв для вирішення даної задачі часто є тільки один шлях, і користувач мимоволі звикає до одного певного способу. Набагато краще мати чотири різних версії застосунку, кожна з яких відповідає певній метафорі користувача інтерфейсу конкретного пристрою, ніж один загальний застосунок, який не може бути інтегрований належним чином ні в один цільовий пристрій.

2.3 Стек MEAN

MEAN стек (MongoDB, Express.js, Angular, Node.js) є популярним набором технологій для створення динамічних веб-застосунків і мобільних застосунків. Його основною перевагою є використання JavaScript на всіх рівнях розробки, що значно спрощує процес створення і підтримки програмного забезпечення [23].

MongoDB – це нереляційна база даних, яка зберігає дані у вигляді документів у форматі BSON (бінарний JSON). MongoDB забезпечує гнучкість та масштабованість завдяки можливості горизонтального шардінгу. Документно-орієнтована модель дозволяє зберігати складні структури даних без необхідності створення складних таблиць і з'єднань, як у традиційних реляційних базах даних. MongoDB підтримує індексацію, агрегації та реплікацію, що підвищує продуктивність і надійність застосунків.

Express.js – це мінімалістичний веб-фреймворк для Node.js, який спрощує розробку серверної частини застосунків. Express.js надає потужні інструменти для обробки HTTP-запитів, маршрутизації, середовища для розробки, тестування і забезпечення безпеки застосунків. Він підтримує безліч проміжних обробників, що дозволяє легко інтегрувати сторонні бібліотеки та розширювати функціональність застосунків.

Angular – це фронтенд-фреймворк, розроблений Google, який використовується для створення динамічних односторінкових застосунків (SPA) [26]. Angular забезпечує двосторонню прив’язку даних, що дозволяє автоматично оновлювати модель і вигляд застосунку при зміні даних. Він також підтримує модульну структуру, що спрощує організацію коду і його повторне використання. Angular включає вбудовані інструменти для тестування, налагодження і оптимізації застосунтків, що підвищує якість і продуктивність розробки.

Node.js – це середовище виконання JavaScript, яке дозволяє запускати код на сервері. Node.js використовує неблокуючу, подієво-орієнтовану архітектуру, що дозволяє обробляти багато одночасних з'єднань з низькою затримкою. Це робить його ідеальним для створення високоефективних серверів і мережевих застосунків. Node.js має велику екосистему модулів і бібліотек, доступних через npm (Node Package Manager), що спрощує розробку і розширення застосунків.

Переваги використання MEAN стеку для розробки мобільних застосунків включають кілька суттєвих аспектів. По-перше, використання єдиного мовного середовища, оскільки всі компоненти MEAN стеку використовують JavaScript, значно спрощує процес розробки та обслуговування застосунків. Розробники можуть писати весь код (клієнтський, серверний і базу даних) однією мовою, що знижує витрати на навчання і підвищує ефективність роботи команди. По-друге, MEAN стек забезпечує повний цикл розробки, включаючи базу даних, серверну логіку, бізнес-логіку і фронтенд, що дозволяє розробникам зосередитися на вирішенні конкретних завдань, не витрачаючи час на інтеграцію різних технологій. Третя перевага полягає в масштабованості кожного компонента MEAN стеку, що дозволяє створювати застосунки, здатні витримувати великий обсяг трафіку і даних. MongoDB добре масштабується горизонтально, а Node.js відомий своєю здатністю обробляти безліч

одночасних з'єднань, що робить його ідеальним для побудови масштабованих застосунків.

Ще однією важливою перевагою є швидкість розробки. Використання єдиного мовного середовища і потужних інструментів Angular для фронтенду значно прискорює процес розробки. Express.js спрощує створення серверної частини, а MongoDB забезпечує гнучке зберігання даних. Крім того, наявність великої кількості готових модулів і бібліотек для Node.js дозволяє швидко реалізовувати різноманітні функціональні можливості. Активна спільнота і підтримка MEAN стеку також є важливим фактором. MEAN стек має активну спільноту розробників, що забезпечує доступ до великої кількості ресурсів, бібліотек, інструментів та документації. Це значно спрощує вирішення проблем і розвиток проекту, оскільки розробники можуть знайти відповіді на свої запитання і приклади реалізації необхідних функцій.

Порівняння MEAN з іншими популярними технологічними стеками допоможе краще зрозуміти його переваги і недоліки. Наприклад, LAMP (Linux, Apache, MySQL, PHP) є традиційним стеком для веб-розробки, який складається з Linux як операційної системи, Apache як веб-сервера, MySQL як бази даних і PHP як серверної мови програмування. LAMP широко використовується завдяки своїй надійності та зрілості. Проте, він вимагає використання різних мов програмування для різних частин застосунку, що може збільшити складність розробки і обслуговування. На відміну від LAMP, MEAN використовує єдину мову програмування (JavaScript) на всіх рівнях, що спрощує розробку і знижує витрати на навчання команди.

Інший популярний стек, MERN (MongoDB, Express.js, React, Node.js), є аналогом MEAN, але замість Angular використовує React для фронтенду. React [21], розроблений Facebook, дозволяє створювати динамічні і високоінтерактивні користувацькі інтерфейси. MERN стек має ті ж переваги, що і MEAN, включаючи єдину мовну платформу і масштабованість. Вибір між Angular і React залежить від конкретних вимог проекту і переваг

розробників. Angular може бути більш підходящим для великих проектів з комплексною архітектурою, тоді як React забезпечує більш гнучкий підхід для створення компонентів інтерфейсу.

Ще один сучасний підхід до розробки вебзастосунків – JAMstack (JavaScript, APIs, Markup), який фокусується на використанні JavaScript для динамічної взаємодії, API для серверної логіки і статичної розмітки для контенту. JAMstack забезпечує високу продуктивність і безпеку, оскільки більшість контенту є статичним. Проте, цей підхід може бути менш гнучким для складних динамічних застосунків. MEAN стек надає більш універсальний підхід для створення динамічних застосунків з повною інтеграцією серверної і клієнтської частин.

Комбінація Django (веб-фреймворк для Python) і React є популярним варіантом для створення сучасних вебзастосунків. Django забезпечує потужні можливості для розробки серверної частини, включаючи вбудовану адмін-панель, ORM і систему аутентифікації. React використовується для створення інтерактивних користувацьких інтерфейсів. Проте, використання двох різних мов програмування (Python і JavaScript) може збільшити складність розробки і потребувати додаткових зусиль для інтеграції. MEAN стек, з єдиною мовною платформою, надає більш уніфікований підхід.

Схожий стек, Spring Boot + Angular, використовує Spring Boot (веб-фреймворк для Java) для серверної частини і Angular для клієнтської. Spring Boot забезпечує високу продуктивність і надійність, що робить його ідеальним для створення великих корпоративних додатків. Angular надає потужні інструменти для створення динамічних інтерфейсів. Однак, використання Java для серверної частини і JavaScript для клієнтської може вимагати додаткових зусиль для навчання і підтримки розробників.

Таким чином, MEAN стек має безліч переваг для розробки мобільних застосунків, включаючи єдину мовну платформу, повний цикл розробки, високу масштабованість, швидкість розробки і активну спільноту. Порівняння з іншими популярними стеками показує, що MEAN забезпечує

більш уніфікований і універсальний підхід до створення динамічних застосунків.

2.4 Вимоги до створюваного програмного продукту

Мобільний застосунок Spike, який розробляється, повинен відповідати ряду технічних вимог для забезпечення його функціональності, надійності, безпеки та зручності використання. Застосунок має підтримувати мультиплатформену роботу, тобто бути доступним для користувачів на операційних системах Android та iOS. Це досягається використанням крос-платформних інструментів розробки, таких як React Native або Flutter, які дозволяють розробляти застосунок одночасно для обох платформ з мінімальними змінами в коді.

Однією з основних вимог є висока продуктивність і швидкість роботи застосунку. Користувачі повинні мати змогу взаємодіяти із застосунком без затримок і зависань. Це включає оптимізацію коду, ефективне використання ресурсів пристрою та мінімізацію навантаження на серверну частину. Наприклад, я використовував методи асинхронного програмування та обробки великих обсягів даних для забезпечення швидкої і надійної роботи застосунку.

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, зручним і привабливим. Дизайн має відповідати сучасним стандартам UX/UI та бути адаптивним, тобто коректно відображатися на різних розмірах екранів і роздільних здатностях. Для цього я використовував бібліотеки компонентів, такі як Material-UI або Bootstrap, що забезпечують високу якість та консистентність дизайну.

Застосунок повинен забезпечувати безперебійний доступ до даних користувачів та їх синхронізацію між різними пристроями. Це включає розробку механізмів для роботи в режимі офлайн з подальшою синхронізацією даних при відновленні підключення до Інтернету. Для цього

я застосував технології, такі як IndexedDB та Service Workers, які дозволяють зберігати дані локально і синхронізувати їх з сервером.

Важливою вимогою є забезпечення високого рівня безпеки даних користувачів. Це включає шифрування даних на пристрої та під час їх передачі, а також реалізацію надійних механізмів аутентифікації та авторизації, таких як двофакторна аутентифікація та використання токенів (JWT). Я реалізував ці механізми, використовуючи бібліотеки, такі як bcrypt для хешування паролів та jsonwebtoken для роботи з токенами.

Застосунок повинен мати модульну архітектуру, що дозволить легко додавати нові функції та оновлювати існуючі без значних змін в коді. Це сприяє підтримці та розвитку застосунку в майбутньому. Для цього я застосував принципи модульного програмування та архітектурні патерни, такі як MVC (Model-View-Controller) або MVVM (Model-View-ViewModel).

Забезпечення можливості інтеграції з іншими сервісами та платформами через API є ще однією важливою вимогою. Це дозволить розширити функціональність застосунку та забезпечити взаємодію з іншими системами. Я використав RESTful API для забезпечення цієї взаємодії, що дозволяє легко інтегрувати застосунок з іншими сервісами.

Застосунок повинен підтримувати локалізацію, тобто можливість відображення інтерфейсу та контенту на різних мовах, що дозволить використовувати його користувачам з різних країн. Для цього я використовував бібліотеки, такі як i18next або React-Intl, які спрощують процес локалізації та міжнародної адаптації застосунку.

Наявність системи аналітики, яка дозволить відстежувати активність користувачів, виявляти проблеми та покращувати функціональність застосунку на основі зібраних даних, є також важливою вимогою. Я інтегрував Google Analytics та Firebase Analytics для забезпечення детального аналізу використання застосунку та збору статистичних даних.

2.5 Вибрані інструментальні засоби

Для розробки мобільного застосунку Spike необхідно використовувати відповідне інструментальне програмне забезпечення, яке забезпечить ефективний процес розробки, тестування та впровадження. Основним інструментом розробки є інтегроване середовище розробки (IDE). Для створення крос-платформного мобільного застосунку я використовував Visual Studio Code, оскільки воно підтримує роботу з React Native та надає всі необхідні інструменти для написання, тестування та налагодження коду.

Для забезпечення контролю версій і співпраці між розробниками використовується система керування версіями Git. Я використовував GitHub для зберігання коду у віддаленому репозиторії, що забезпечує зручний доступ до коду, можливість відстеження змін та спільної роботи над проектом.

Важливою частиною розробки є налаштування CI/CD (Continuous Integration/Continuous Deployment) процесів. Це забезпечує автоматичне тестування та розгортання застосунку. Для цього я використовував GitHub Actions, що дозволяє налаштувати автоматичне виконання тестів при кожному коміті та автоматичне розгортання нових версій застосунку.

Для розробки серверної частини застосунку використовуються Node.js та Express.js. Це забезпечує високу продуктивність та масштабованість серверної частини. Для роботи з базами даних я налаштував MongoDB, яка забезпечує швидкий доступ до даних та їх гнучке зберігання. Інструменти для управління базою даних, такі як MongoDB Compass, дозволяють ефективно працювати з даними під час розробки та тестування.

Для забезпечення безпеки та управління користувацькими сесіями використовуються бібліотеки для аутентифікації та авторизації, такі як Passport.js. Це забезпечує реалізацію різних методів аутентифікації та управління токенами, що підвищує безпеку застосунку.

Для написання та виконання тестів використовуються фреймворки для тестування, такі як Jest та React Testing Library для клієнтської частини, і Mocha для серверної частини. Це дозволяє забезпечити високу якість коду та виявляти помилки на ранніх етапах розробки.

Нарешті, для забезпечення моніторингу та аналітики роботи застосунку використовуються інструменти, такі як Google Analytics та Firebase Analytics. Вони дозволяють відстежувати активність користувачів, збирати статистичні дані та аналізувати поведінку користувачів для подальшого вдосконалення застосунку.

Таким чином, дотримання вимог до програмного продукту та використання відповідного інструментального програмного забезпечення є ключовими аспектами успішної розробки, впровадження та підтримки мобільного застосунку Spike.

РОЗДІЛ 3
РОЗРОБКА І ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Призначення мобільного застосунку. Технічне завдання

Перед розробкою застосунку було поставлено такі завдання:

Застосунок повинен володіти такими особливостями:

– Гнучкістю, зручною для адміністраторів системою управління структурою.

– Для користувачів застосунку повинен бути створений розділ перегляду усіх можливих функцій.

– Застосунок повинен бути розроблений з максимально простим інтерфейсом.

– Застосунок повинен бути розроблений на системі управління змістом, яка б дозволяла вносити зміни в нього, з можливістю розмежування прав доступу до вмісту і незалежністю від технічних фахівців.

Програма повинна бути складена з двох частин − серверної, яку представляє “backend” за допомогою власного серверу, та клієнтської, на базі операційної системи, яка знаходиться на мобільному пристрої користувача. Серверна частина повинна містити перелік користувачів з даними для входу, запрограмований об’єм даних з запитами на їх отримання, та додання даних з клієнта.

Вдалий мобільний застосунок – це надзвичайно ефективний інструмент торгівлі – він здатний захоплювати увагу аудиторії. Як і будь-який інший маркетинговий інструмент, заснований на принципі безпосереднього відгуку, перш за все він повинен заінтригувати відвідувача, а потім спонукати його на певні дії.

Багато хто ігнорує ці особливості головної сторінки, що часто призводить до того, що відвідувачі не затримуються у застосунку надовго і залишають його, ледь зайшовши. Такі рішення, нехай навіть і містять іноді

величезну кількість корисних порад та статей та багато інших функцій, але практично ніколи не досягають передбачуваного рівня відвідуваності, не кажучи вже про продажі.

Мобільний застосунок в гостьовому режимі має такі сторінки:

– *Авторизація*. Дозволяє користувачу авторизуватись в системі за допомогою існуючого логіну та паролю.

– *Реєстрація*. Існує для створення нового користувача в базі даних застосунку.

– *Відновлення паролю*. Якщо користувач забув пароль, то на цій сторінці він зможе відновити його за допомогою свого e-mail адресу.

Після авторизації для користувача відкривається доступ до таких сторінок:

– Сторінка, де знаходиться місячний бюджет,витрати та інша статистика.
– Сторінка «Статок» , де знаходяться всі створенні гаманці користувача.
– Сторінка «Доходи та витрати», де знаходиться загальна статистика користувача.
– Сторінка «Налаштування».

Обрано назву застосунку – «**Spike**». Його сфера застосування має охоплювати широкий спектр фінансових аспектів, від щоденного обліку витрат до довгострокового фінансового планування. Наприклад :

1. *Особисте фінансове планування*

Spike надає користувачам можливість створювати особисті бюджети, визначати фінансові цілі та планувати витрати. Це допомагає користувачам краще розуміти свої фінансові можливості та уникати перевитрат.

2. *Контроль за витратами*

Застосунок дозволяє користувачам фіксувати всі свої витрати, розподіляючи їх по категоріях. Це надає можливість аналізувати свої витрати, виявляти основні джерела витрат та оптимізувати їх для досягнення фінансової стабільності.

3. *Заощадження*

Spike допомагає користувачам відкладати гроші на майбутні потреби. Застосунок може рекомендувати оптимальні способи заощадження.

4. *Фінансова грамотність*

Spike також містить освітні матеріали та поради щодо фінансового управління, що допомагає користувачам покращувати свої знання в галузі фінансів та приймати обдумані фінансові рішення.

5. *Сімейний бюджет*

Застосунок дозволяє вести спільний облік фінансів для родин, що допомагає ефективно планувати сімейний бюджет, координувати витрати та досягати спільних фінансових цілей.

6. *Мобільність та доступність*

Spike доступний на мобільних пристроях, що дозволяє користувачам керувати своїми фінансами в будь-який час і в будь-якому місці. Це робить управління фінансами більш зручним та доступним.

Розробка мобільного застосунку Spike спрямована на створення зручного та ефективного інструменту для управління особистими фінансами. Цей застосунок має допомагати користувачам контролювати свої доходи та витрати, планувати бюджет, заощаджувати кошти та досягати фінансових цілей. Розробка Spike повинна допомогти вирішити кілька ключових завдань. Зокрема, він повинен допомагати користувачам відстежувати свої фінансові транзакції, фіксуючи доходи та витрати. Це корисно для отримання чіткого уявлення про фінансовий стан, щоб на основі цього приймати обґрунтовані фінансові рішення, це є важливим аспектом для досягнення фінансової стабільності та уникнення боргів.

Метою розробки є також створення інструменту, який допоможе користувачам складати та дотримуватися бюджету. Spike дозволяє розподіляти доходи на різні категорії витрат, встановлювати ліміти та стежити за їх дотриманням. Це сприяє раціональному використанню коштів та запобіганню перевитратам. Застосунок надає інструменти для відстеження

прогресу досягнення цих цілей, що сприяє їх успішній реалізації. Одним з важливих аспектів призначення розробки є стимулювання користувачів до заощадження коштів. Завдяки цьому мобільному застосунку користувачі можуть ефективно контролювати свої фінанси, планувати витрати та доходи, а також досягати фінансових цілей, що в результаті сприяє підвищенню фінансової грамотності та добробуту.

3.2 Етапи розробки застосунку та його функціональні можливості

Стадії та етапи розробки

Стадія перша: Розпочинається створення мобільного застосунку "Spike" з визначення його основних вимог та функціональності. На цьому етапі проводиться аналіз потреб користувачів та визначення цілей застосунку.

Стадія друга: Розробляється дизайн інтерфейсу застосунку з використанням програмного забезпечення для дизайну, наприклад, "Figma". Тут створюються макети та ілюстрації, які відображають вигляд та функціональність кожної сторінки застосунку.

Стадія третя: Після розробки дизайну створюються всі необхідні сторінки та налаштовується маршрутизація у застосунку. Це дозволяє користувачам отримати доступ до різних частин застосунку за допомогою посилань.

Стадія четверта: Розробляється і налаштовується REST API для взаємодії клієнтської та серверної частин застосунку. Тут створюються необхідні схеми та функціонал для обробки запитів користувачів.

Стадія п'ята: Продовжується розробка та вдосконалення програмного продукту з боку клієнтської частини застосунку. Виконуються роботи з додавання нових функцій та виправлення виявлених помилок.

Стадія шоста: Проводиться тестування застосунку з метою виявлення та усунення будь-яких недоліків, помилок і проблем в його роботі.

Тестування виконується для забезпечення якості та надійності програмного продукту.

Стадія сьома: Після виявлення недоліків та помилок, їхнє усунення та виправлення виконується у відповідності до вимог і відповідно до плану. Всі виявлені проблеми та завдання для виправлення фіксуються і вирішуються.

Стадія восьма: Після усунення недоліків та вирішення всіх задач розробка застосунку завершується, і програмний продукт готовий до випуску на ринок або використання користувачами.

Функціональні можливості застосунку

Розроблюваний мобільний застосунок потрібен для управління особистими фінансами, він покликаний надавати користувачам зручні інструменти для ведення, планування та контролю їх бюджету. Він інтегрує в собі широкий спектр функціональних можливостей, спрямованих на підтримку фінансової дисципліни та ефективного управління фінансами.

Основні функції застосунку включають авторизацію, що дозволяє користувачам входити в систему за допомогою облікових даних (логін та пароль). Реєстрація нових користувачів здійснюється шляхом введення особистих даних, таких як ім'я, прізвище, e-mail та пароль. У разі забуття паролю, користувачі можуть відновити доступ до свого облікового запису, отримавши інструкції на вказаний e-mail.

Застосунок дозволяє встановлювати місячний бюджет і моніторити його виконання за допомогою деталізованих графіків та діаграм, які ілюструють розподіл витрат за різними категоріями. Функція "Статок" надає можливість ведення обліку всіх фінансових рахунків і гаманців користувача, включаючи банківські рахунки, електронні гаманці та готівку.

Управління доходами та витратами у застосунку реалізоване через можливість створення та редагування категорій доходів і витрат. Користувачі можуть додавати нові транзакції з вказанням опису, суми та категорії для кожної операції. Звіти та аналітика, які генерує застосунок, дозволяють користувачам аналізувати свої фінанси за різними періодами,

використовуючи графіки та діаграми для наглядного представлення інформації.

Планування бюджету та витрат реалізоване за допомогою “підсказок”, які користувач може переглядати у будь-який час.

У розділі налаштувань користувачі можуть персоналізувати застосунок під свої потреби. Захист даних забезпечується за допомогою шифрування і двофакторної автентифікації, що забезпечує високий рівень безпеки інформації користувача.

Застосунок підтримує синхронізацію даних між різними пристроями, що дозволяє користувачам мати доступ до своїх фінансових даних з будь-якого місця.

Додаткові функції включають додавання банківської карти, перегляд політики конфідеційності, умов використання та перегляд більш розгорнутих графіків та статистики.

3.3 Серверна архітектура застосунку

Архітектура клієнт-серверу є одним із архітектурних шаблонів програмного забезпечення, який відіграє важливу роль у створенні розподілених мережних застосунків. Вона забезпечує взаємодію та обмін даними між клієнтом та сервером, що є ключовим для функціонування мобільного застосунку Spike. Основні компоненти архітектури клієнт-серверу включають набір серверів, які надають інформацію та сервіси, до яких звертаються клієнти; набір клієнтів, які використовують сервіси, що надаються серверами; мережу, яка забезпечує взаємодію між клієнтами та серверами.

У мобільному застосунку Spike використовується REST (Representational State Transfer) – підхід до архітектури мережевих протоколів, який забезпечує доступ до інформаційних ресурсів. REST базується на принципах функціонування Всесвітньої павутини, зокрема

можливостях HTTP. Відповідно до принципів REST, дані повинні передаватися у вигляді стандартних форматів, таких як HTML, XML, або JSON. REST протоколи, включаючи HTTP, підтримують кешування, не залежать від мережевого прошарку і не зберігають інформацію про стан між парами «запит-відповідь». Такий підхід забезпечує масштабованість системи і дозволяє їй адаптуватися до нових вимог.

Клієнт-серверні застосунки, включаючи Spike, є одними з найпоширеніших і водночас найскладніших у розробці. Дії, які можна виконувати над ресурсом, визначаються стандартними повідомленнями протоколу. У системі WWW цей протокол – HTTP, але існують REST-архітектури, що використовують й інші протоколи [22].

Найчастіше у мобільному застосунку Spike використовуються чотири типи запитів:

- GET – отримати ресурс. Цей запит використовується для отримання даних з сервера, таких як інформація про доходи та витрати користувача.
- POST – створити новий ресурс. Використовується для додавання нових фінансових записів або цілей користувача.
- PUT – замінити стан поточного ресурсу. Дозволяє оновлювати існуючі дані, наприклад, редагувати інформацію про витрати.
- DELETE – видалити ресурс. Забезпечує видалення фінансових записів або інших даних з бази.

Ця архітектура забезпечує гнучкість та ефективність мобільного застосунку Spike, дозволяючи користувачам зручно керувати своїми фінансами, отримуючи доступ до необхідної інформації та здійснюючи необхідні дії з будь-якого місця та в будь-який час. Вибір REST-архітектури дозволяє застосунку легко масштабуватися та адаптуватися до нових вимог, забезпечуючи стабільну і надійну роботу навіть при значному збільшенні кількості користувачів або обсягу оброблюваних даних.

3.4 Аутентифікація та авторизація

Аутентифікація та авторизація користувачів є критичними аспектами мобільного застосунку Spike, що забезпечують безпеку та конфіденційність даних користувачів. Цей розділ детально описує основні методи аутентифікації, використання токенів для забезпечення безпеки, а також налаштування прав доступу до різних функцій та даних застосунку. Аутентифікація є першим кроком у процесі забезпечення безпеки користувацьких даних. У мобільному застосунку Spike використовуються кілька методів аутентифікації, які забезпечують різні рівні безпеки та зручності для користувачів.

Парольна аутентифікація є найпоширенішим методом, при якому користувач вводить свій логін та пароль. Система перевіряє ці дані і надає доступ, якщо вони правильні. Для підвищення безпеки використовується шифрування паролів, наприклад, через bcrypt, і зберігання лише хешованих версій паролів у базі даних. Для забезпечення стійкості паролів до атак, користувачам рекомендується створювати паролі з великої кількості символів, використовуючи комбінації літер, цифр і спеціальних символів. Двофакторна аутентифікація (2FA) використовується для додаткового рівня безпеки. Вона вимагає введення одноразового коду, надісланого на телефон або електронну пошту користувача, після введення правильного пароля. Це значно ускладнює доступ до акаунту неавторизованим особам. Соціальна аутентифікація дозволяє користувачам входити в систему через свої облікові записи в соціальних мережах, таких як Google або Facebook. Це спрощує процес входу для користувачів і забезпечує додатковий рівень перевірки, оскільки соціальні платформи мають власні механізми безпеки. Біометрична аутентифікація використовує біометричні дані, такі як відбитки пальців або розпізнавання обличчя, для аутентифікації користувачів. Цей метод забезпечує високий рівень безпеки і зручності, оскільки біометричні дані важко підробити або вкрасти.

Для забезпечення безпеки та управління сесіями в Spike використовується методика аутентифікації на основі токенів, зокрема, JSON Web Tokens (JWT) [29]. Використання токенів забезпечує декілька переваг. Безпека передачі даних забезпечується шляхом передачі токенів між клієнтом і сервером через захищені канали (HTTPS), що мінімізує ризик перехоплення даних. Всі дані, які передаються через HTTP, шифруються, що забезпечує конфіденційність і цілісність переданої інформації. Структура токенів складається з трьох частин – заголовка, корисного навантаження та підпису. Корисне навантаження містить інформацію про користувача і термін дії токена, а підпис забезпечує його цілісність і захист від підробки. Заголовок містить інформацію про алгоритм шифрування, використаний для створення підпису. Підпис забезпечує цілісність токена, захищаючи його від маніпуляцій. Після успішної аутентифікації сервер генерує токен і передає його клієнту. Клієнт зберігає токен, наприклад, в локальному сховищі, і використовує його для авторизації наступних запитів до сервера. Це дозволяє користувачам залишатися авторизованими протягом тривалого часу без необхідності повторної аутентифікації. Токени мають обмежений термін дії, після закінчення якого користувач повинен пройти аутентифікацію знову. Це знижує ризик використання вкрадених токенів. Термін дії токена може бути налаштований відповідно до вимог безпеки конкретного застосунку. Наприклад, для високобезпекових застосунків термін дії токена може бути коротшим. Для забезпечення безперебійної роботи і покращення користувацького досвіду застосовуються механізми оновлення токенів (refresh tokens), які дозволяють продовжити сесію без повторної аутентифікації.

Налаштування прав доступу в Spike забезпечує, що кожен користувач має доступ лише до тих функцій і даних, які їм дозволено. Це досягається шляхом реалізації ролей і політик доступу. Користувачі можуть мати різні ролі, наприклад, адміністратор або звичайний користувач, які визначають рівень їхнього доступу. Адміністратори мають повний доступ до всіх

функцій та даних, тоді як звичайні користувачі мають обмежений доступ. Кожна роль має чітко визначений набір прав, що дозволяє точно контролювати доступ до різних частин застосунку. Встановлюються правила, які визначають, які дії може виконувати користувач з певною роллю. Наприклад, звичайні користувачі можуть мати доступ до перегляду та редагування своїх фінансових записів, але не можуть видаляти інші облікові записи. Політики доступу також можуть визначати, які частини інтерфейсу користувача будуть доступні для конкретної ролі. Кожен запит до сервера перевіряється на відповідність політикам доступу. Якщо користувач не має необхідних прав, сервер відхиляє запит і повертає відповідне повідомлення про помилку. Це забезпечує захист даних і функцій від несанкціонованого доступу. Контроль доступу на рівні API також дозволяє реалізувати більш гнучкі та складні політики доступу, враховуючи різні умови та контексти використання. Для забезпечення додаткової безпеки і відстеження підозрілої активності, Spike веде логування всіх спроб доступу до даних і функцій. Це дозволяє адміністраторам вчасно виявляти та реагувати на потенційні загрози безпеці. Для покращення гнучкості та адаптивності системи, права доступу можуть динамічно змінюватися в залежності від поведінки користувача та контексту його дій. Наприклад, система може тимчасово підвищити права доступу користувача при виконанні певних завдань або обмежити доступ у разі виявлення підозрілої активності.

Таким чином, комплексна система аутентифікації та авторизації в мобільному застосунку Spike забезпечує високий рівень безпеки даних користувачів, захист від несанкціонованого доступу, а також зручність і гнучкість в управлінні правами доступу. Це дозволяє користувачам безпечно та ефективно користуватися всіма функціями застосунка, зберігаючи при цьому високий рівень захисту особистої інформації.

3.5 Архітектура інтерфейсу мобільного застосунку

Мобільний застосунок для управління особистими фінансами має чітко структурований інтерфейс, який забезпечує зручний доступ до всіх необхідних функцій. Архітектура інтерфейсу складається з кількох основних сторінок, кожна з яких виконує специфічні завдання.

Авторизація

Сторінка авторизації дозволяє користувачам увійти в систему за допомогою існуючого логіну та паролю (рис.3.1). Вона включає наступні елементи:

- Поле для введення логіну.
- Поле для введення паролю.
- Кнопка для входу в систему.
- Кнопка для відновлення паролю
- Кнопка реєстрації

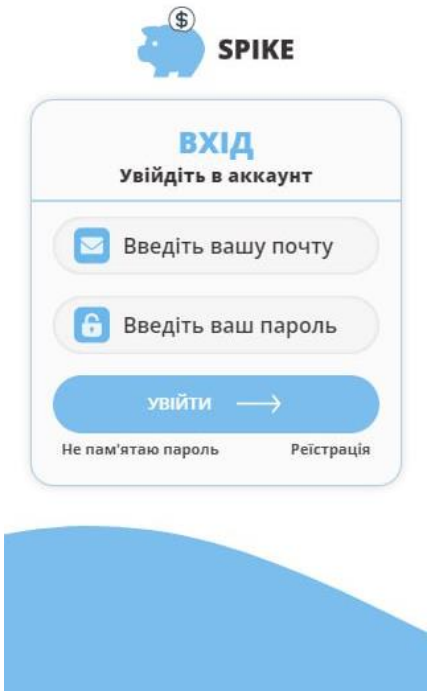

Рис. 3.1 – Авторизація

Реєстрація

Сторінка реєстрації призначена для створення нового облікового запису в системі (рис.3.2). Вона містить:

- Поля для введення особистих даних (ім'я, e-mail, пароль).
- Кнопку для завершення реєстрації.

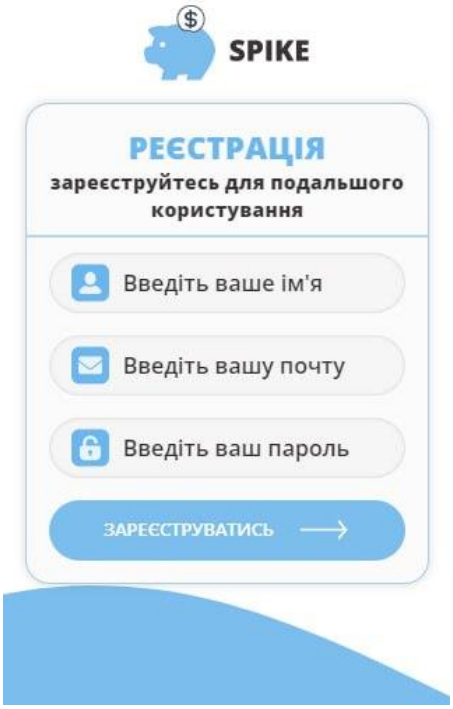

Рис. 3.2 – Реєстрація

Відновлення паролю

Ця сторінка надає можливість відновити забутий пароль (рис.3.3-3.4).

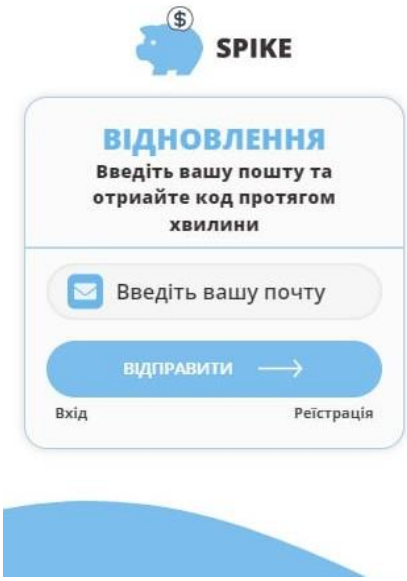

Рис. 3.3 – Відновлення паролю

Рис. 3.4 – Введення коду

Основні елементи сторінки:

- Поле для введення e-mail адреси, пов'язаної з обліковим записом.
- Кнопка для надсилання коду на вказану адресу для відновлення паролю.

Головні сторінки після авторизації

Після успішної авторизації користувачу відкривається доступ до основних функцій застосунку через наступні сторінки (рис.3.5).

Місячний бюджет

Ця сторінка містить інформацію про місячний бюджет користувача, витрати та іншу фінансову статистику. Вона включає:

- Місячний бюджет.
- Графік, що ілюструє витрати за місяць.

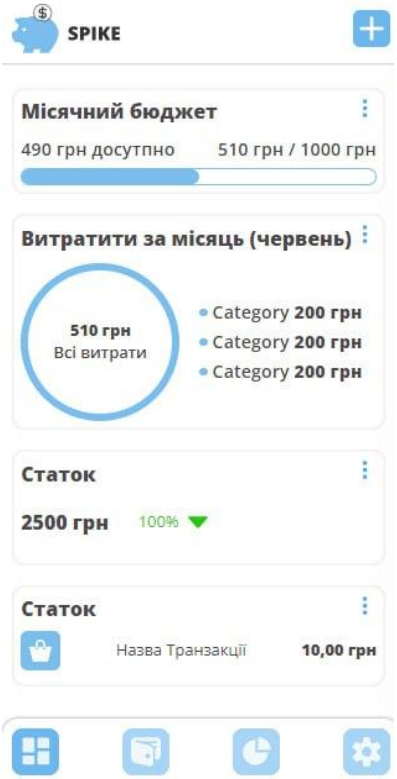

Рис. 3.5 – Головна сторінка

Статок

На цій сторінці знаходяться всі створені гаманці користувача (рис.3.6).

Основні компоненти:

- Список гаманців із зазначенням балансу кожного.
- Кнопки для додавання, редагування та видалення гаманців.
- Інформація про останні транзакції по кожному гаманцю.

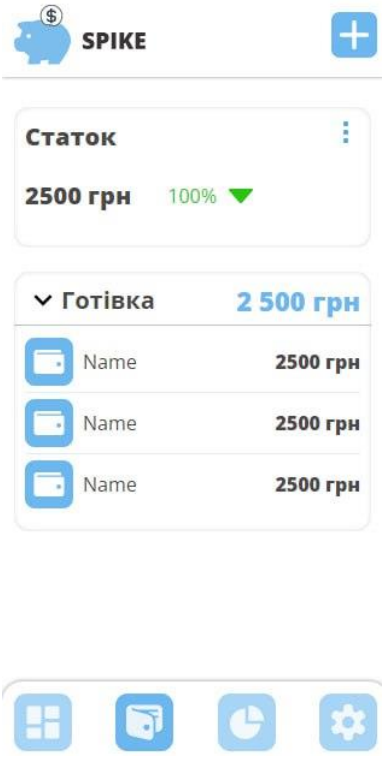

Рис. 3.6 – Статок

Доходи та витрати

Сторінка забезпечує загальну статистику фінансових операцій користувача (рис.3.7).

Вона містить:

- Графіки доходів та витрат за певні періоди.
- Загальний бюджет.
- Динаміку витрат.
- Розділ “Інше”, який містить сторінку “Статок” та “Історія транзакцій”

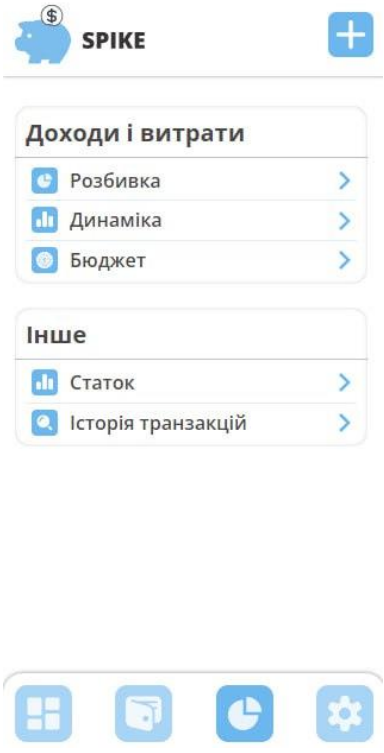

Рис. 3.7 – Доходи та витрати

Налаштування

Сторінка налаштувань дозволяє користувачу персоналізувати застосунок (рис.3.8). Вона включає:

- Налаштування фінансових просторів.
- Підключення банківської карти (рис.3.9).
- Зміна зовнішнього виноряду застосунку, а саме зміна “Теми” застосунку.
- Політику конфіденційності.
- Умови використання.

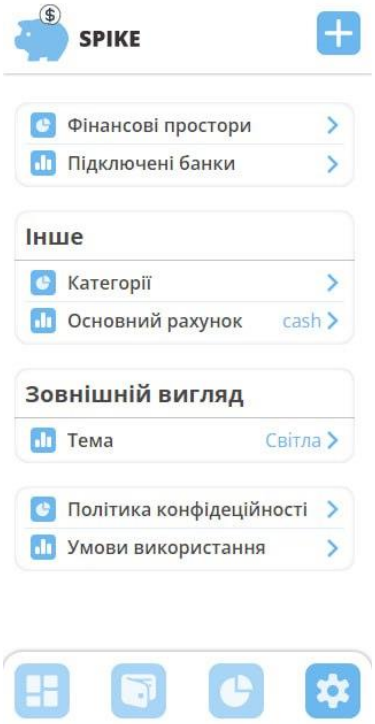

Рис. 3.8 – Налаштування

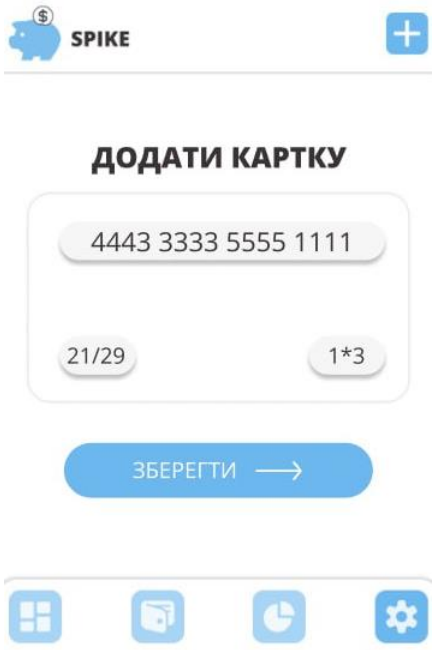

Рис. 3.9 – Додавання карти

3.6 Види тестування застосунку

Методи тестування включали не лише функціональні, але й нефункціональні аспекти. Функціональне тестування перевіряло, чи виконує застосунок свої основні функції відповідно до вимог. Нефункціональне тестування охоплювало продуктивність, безпеку, зручність використання та інші характеристики, які не пов'язані безпосередньо з основними функціями, але є критичними для успіху застосунку.

Під час тестування особливу увагу слід було виділено тестуванню продуктивності, що включало перевірку швидкості роботи застосунку під різним навантаженням. Це допомогло виявити можливі проблеми з продуктивністю, які можуть виникати при високому навантаженні на систему. Тестування безпеки забезпечувало захист даних користувачів та запобігало несанкціонованому доступу до інформації.

Для проведення тестування я використовував різні інструменти та методи, такі як автоматизоване тестування, яке значно прискорювало процес перевірки застосунку. Автоматизоване тестування дозволило створювати тести, які виконувалися без втручання людини, що забезпечувало постійну перевірку коректності роботи застосунку при кожній зміні в коді.

Регресійне тестування було ще одним важливим видом тестування, яке перевіряло, чи не призвели нові зміни у коді до появи нових помилок у вже функціонуючих частинах застосунку. Це дозволяло вчасно виявити та усунути нові проблеми, що виникають після внесення змін до коду.

Інтеграційне тестування перевіряло коректну роботу застосунку у поєднанні з іншими системами та модулями. Це включало тестування взаємодії між різними компонентами застосунку, що дозволяло переконатися у їхній сумісності та коректній взаємодії.

Системне тестування охоплювало перевірку всього застосунку в цілому, включаючи всі його компоненти та модулі. Це дозволяло переконатися, що всі частини застосунку працюють разом коректно та відповідають вимогам.

Окрім основного тестування, було також проведено користувацьке тестування, яке включало тестування застосунку реальними користувачами. Це допомагало отримати зворотний зв'язок від кінцевих користувачів та виявити проблеми, які можуть виникати під час реального використання застосунку.

Підтримка застосунку після його впровадження включала регулярні оновлення та виправлення помилок. Це забезпечувало актуальність застосунку та його відповідність змінним вимогам користувачів. Регулярні оновлення також включали додавання нових функцій та поліпшення існуючих, що підвищувало привабливість застосунку для користувачів.

Моніторинг роботи застосунку був важливим аспектом підтримки, що включав спостереження за його продуктивністю та стабільністю. Це дозволяло вчасно виявляти та усувати проблеми, що виникають під час експлуатації, забезпечуючи безперебійну роботу застосунку Spike.

ВИСНОВКИ

В процесі підготовки кваліфікаційної роботи бакалавра було досягнуто поставлену мету і виконані всі завдання дослідження. Зокрема, розроблено мобільний застосунок Spike, який призначений для ефективного управління особистими фінансами користувачів. Цей застосунок забезпечує широкий спектр функціональних можливостей, які дозволяють користувачам вести облік доходів та витрат, планувати фінансові цілі, аналізувати фінансові звіти та отримувати корисні рекомендації щодо заощаджень та інвестування.

На початковому етапі роботи було проведено аналіз існуючих рішень у сфері мобільних фінансових застосунків, що дозволило визначити ключові вимоги до розроблюваного продукту. Основну увагу було приділено аналізу зручності користувацького інтерфейсу, дослідженню інструментів безпеки даних та гнучкості налаштувань.

Для реалізації застосунку було обрано MEAN стек технологій (MongoDB, Express.js, Angular, Node.js), що забезпечило високу продуктивність, масштабованість та гнучкість системи. Під час розробки було сформовано детальну структуру бази даних, створено серверну частину (back-end) та клієнтську частину (front-end) застосунку.

Розробка застосунку пройшла кілька етапів, включаючи аналіз вимог, проєктування, програмування, тестування та перший етап впровадження. Особливу увагу було приділено тестуванню застосунку для забезпечення його надійності та стабільності. Було використано різні методи тестування, включаючи функціональне, нефункціональне, тестування безпеки та продуктивності.

Після завершення етапу розробки було проведено його пілотне тестування серед обмеженої групи користувачів. Результати тестування показали високу ефективність та зручність використання застосунку, а також підтвердили доцільність його подальшого розвитку та вдосконалення.

Окрім технічних аспектів, в роботі було розглянуто економічну доцільність створення та впровадження застосунку. Було проаналізовано переваги, які застосунок приносить користувачам та потенційним рекламодавцям. Залучення нової аудиторії, підвищення фінансової грамотності населення та можливість монетизації через цільову рекламу є значними факторами економічного ефекту від використання застосунку.

На основі проведеної роботи можна зробити висновок, що розроблений мобільний застосунок Spike є ефективним інструментом для управління особистими фінансами, який може значно спростити користувачам процес ведення фінансового обліку та планування. Подальший розвиток застосунку включає додавання нових функціональних можливостей, розширення бази користувачів та інтеграцію з іншими фінансовими сервісами.

Результати роботи можуть бути корисними для подальших досліджень та розробок у сфері мобільних фінансових технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adams G. The Complete Guide to Personal Finance: For Teenagers and College Students. New York: Simon & Schuster, 2017. 288 p.

2. Angular Official Documentation. URL : https://angular.io/docs

3. DZone: MEAN Stack Articles.URL : https://dzone.com/articles/mean-stack

4. Express.js Guide. URL : https://expressjs.com

5. Kiyosaki R. Rich Dad Poor Dad: What the Rich Teach Their Kids About Money That the Poor and Middle Class Do Not! -Scottsdale: Plata Publishing, 2017. 336 p.

6. Loudenback T. 10 Life-Changing money lessons from the millionaire URL : https://www.tanzaloudenback.com

7. MEAN Stack Tutorial: MongoDB, ExpressJS, AngularJS and Node.js. URL : https://www.tutorialspoint.com/meanjs

8. Martinez E. The Role of Mobile Technologies in Financial Management. Madrid: Prentice Hall, 2020. 141 p.

9. Node.js Documentation. URL : https://nodejs.org/en/docs

10.Next Door'.-BusinessInsider,2019.URL: https://www.businessinsider.com/personal-finance

11.Freeman, A. Pro Angular 9: Build Powerful and Dynamic Web Apps. Apress, 2020. 809 p.

12.GitHub: MEAN Stack Example Projects. URL : https://github.com/topics/mean-stack

13.Personal Finance."Investopedia.URL: https://www.investopedia.com/terms/p/personalfinance.asp

14.Stack Overflow. Форум для обговорення технічних питань, пов'язаних з розробкою мобільних додатків і MEAN стеком. URL: https://stackoverflow.com

15.The Balance : Personal Finance Management. URL : https://www.thebalance.com/personal-finance-4074039

16.Tech Crunch : Financial Technology News. URL : https://techcrunch.com/tag/fintech

17.Wilson S. Development of Mobile Applications: Strategies and Technologies. London: Pearson Education, 2019. 257 p.

18.Жукова О. М. Фінансове планування та контроль. Київ: Видавництво Україна, 2016.

19.Захарченко І. В. Інвестиційні стратегії та їх роль у фінансовому плануванні. Київ: Наукове товариство імені Шевченка, 2015.

20.Іванова Н. Л. Методологія бюджетування і управління фінансами. Київ: Фінанси і статистика, 2019.

21.Ковальова М. Особисте фінансове планування: принципи та методи. Львів: Світ, 2018.

22.Лисенко В. П. Моделі управління фінансами. Харків: Право, 2017.

23.Осипова С. М. Фінансовий аналіз в управлінні. Донецьк: Видавництво ДонНТУ, 2017.

24. Продоляк Б. Розробка мобільного застосунку для контролю та планування особистого бюджету на стеку MEAN. Збірник матеріалів наукової конференції за підсумками науково-дослідної роботи здобувачів вищої освіти фізико-математичного факультету Кам'янець-Подільського національного університету імені Івана Огієнка у 2023-2024 н.р., 9-10 квітня 2024 року. Кам'янець-Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, фізикоматематичний факультет, 2024. С. 78-80. URL : http://elar.kpnu.edu.ua/xmlui/handle/123456789/8094

25.Романенко А. К. Фінансовий менеджмент і бюджетування. Херсон: Практика, 2019.

26.Соловйова Л. І. Фінансові технології: інновації та перспективи. Київ: Видавництво Національного університету "Києво-Могилянська академія", 2018.

27.Федоров Ю. С. Моделі ефективного управління фінансами. Київ: Логос, 2016.

28.Хоменко О. І. Інвестиційний менеджмент: стратегії і методи. Дніпро: Видавництво Дніпропетровського університету, 2017.

29.Цепов С. Основи фінансового планування. Харків: Видавництво ХНУ, 2019.

30.Черній В. В. Фінансовий аналіз і планування. Київ: КНЕУ, 2018.

31.Шевченко П. І. Управлінський облік і аналіз. Київ: Видавництво КНЕУ, 2020.

32.Мобільні додатки для управління фінансами URL: https://www.epravda.com.ua/publications/2023/02/22/697326

ДОДАТКИ

Додаток А

Основи впровадження застосунку

Впровадження, тестування та підтримка мобільного застосунку Spike є важливими етапами в життєвому циклі його розробки.

Незважаючи на значний прогрес у розвитку мобільних технологій, продуктивність мобільних процесорів все ще поступається серверним. Тому необхідним було ретельне тестування та налагодження застосунку для забезпечення його коректної роботи. Використання режиму "Debugger" стало важливою складовою процесу налагодження, оскільки він допомагав виявити та виправити помилки в коді. Доступ до меню розробника надав додаткові можливості для налагодження, такі як перегляд журналів помилок та попереджень, що з'являються у застосунку. Це дозволило швидко реагувати на проблеми та усувати їх до того, як застосунок буде випущений для кінцевих користувачів. Для доступу до меню розробника я використовував комбінації клавіш або спеціальні команди, що значно полегшувало процес налагодження.

Етап тестування став ключовим у процесі розробки програмного забезпечення, оскільки він допомагав виявляти та виправляти несправності, зменшуючи ризик виникнення помилок під час експлуатації. Тестування включало кілька методів, таких як аналіз граничних значень та тестування методом чорного ящика. Аналіз граничних значень передбачав перевірку поведінки продукту на крайніх значеннях вхідних даних, оскільки саме в цих умовах найчастіше виникають збої. Наприклад, якщо вхідні значення повинні бути в інтервалі [-1.0; 1.0], то необхідно було перевірити роботу програми при значеннях -1.0, 1.0, -1.001 та 1.001 [28].

Тестування методом чорного ящика полягало у перевірці відповідності поведінки системи вимогам без доступу до внутрішньої структури коду. Я працював із системою, подаючи на її входи зовнішні впливи та спостерігаючи за результатами. Це дозволило перевірити правильність

поведінки системи та її реакцію на критичні ситуації, такі як введення некоректних даних.

У ході розробки також застосовувався метод black-box, який дозволив багатьом користувачам, не знайомим з внутрішньою структурою системи, тестувати її, забезпечуючи вхідні дані та перевіряючи результати на основі очікуваних результатів.

Під час розробки застосунку було важливо враховувати рівень заряду акумулятора пристрою, оскільки мобільні застосунки можуть швидко знижувати його. Наприклад, якщо застосунок використовує геозонування, постійне обчислення місцезнаходження користувача може значно впливати на заряд акумулятора. Для оптимізації продуктивності та зменшення споживання енергії було ретельно вибрано підходи до вибору інструментів тестування. Існує компроміс між швидкістю ітерацій та реалістичністю середовища: одні інструменти забезпечують швидкий цикл "вніс зміни – побачив результат", але не моделюють поведінку браузера в точності, тоді як інші можуть використовувати реальне середовище браузера, але знижувати швидкість ітерацій.

Перший етап впровадження мобільного застосунку Spike завершився стадією підтримки, яка включала постійний моніторинг його роботи, оновлення та виправлення помилок. Підтримка також передбачала зворотний зв'язок від користувачів, що дозволяло вчасно реагувати на їхні потреби та проблеми, забезпечуючи постійне покращення продукту. В цілому, впровадження, тестування та підтримка мобільного застосунку Spike були комплексним процесом, що вимагав уваги до деталей, ретельного планування та постійного вдосконалення для забезпечення його успішної роботи та задоволення користувачів.

Важливим аспектом впровадження стала інтеграція з зовнішніми системами та сервісами. Це включало інтеграцію з API (Application Programming Interface) інших систем, що дозволяло обмінюватися даними між застосунками. Наприклад, використання REST API дозволило застосунку взаємодіяти з сервером для отримання та збереження даних, що є критично важливим для багатьох мобільних застосунків.

Додаток Б

Важливі техніко-економічні показники застосунку

Створення нового програмного продукту, такого як мобільний застосунок Spike, передбачає детальний розрахунок економічного ефекту від його функціонування. Проведемо обґрунтування економічної доцільності розробки та впровадження застосунку, розглядаючи ключові переваги та можливості, які він забезпечує.

Мобільний застосунок Spike має на меті:

- Залучення нової аудиторії, яка зацікавлена в ефективному управлінні особистими фінансами. Використання цього застосунку дозволить користувачам отримувати актуальну та корисну інформацію про способи заощадження, інвестування та фінансового планування. Це сприятиме збільшенню бази користувачів та підвищенню їхньої фінансової грамотності.
- Розширення інформаційного поля навколо управління особистими фінансами є важливою перевагою мобільного застосунку Spike. Застосунок надаватиме користувачам різноманітні статті, поради та новини, що стосуються фінансового планування, заощаджень та інвестицій. Це сприятиме підвищенню рівня обізнаності користувачів щодо фінансових питань, допомагаючи їм приймати обґрунтовані рішення.
- Мобільний застосунок є ефективним інструментом для реклами фінансових продуктів та послуг. За допомогою застосунку можна буде розміщувати рекламні оголошення про банківські продукти, страхові поліси, інвестиційні можливості тощо. Це забезпечить додатковий дохід від реклами та сприятиме розвитку партнерських відносин з фінансовими установами.

Однією з важливих переваг мобільного застосунку Spike є підвищення фінансової грамотності населення. Завдяки інтеграції з різноманітними освітніми матеріалами, курсами та вебінарами, користувачі зможуть отримувати знання про основні принципи фінансового планування, методи

заощадження та інвестування. Це сприятиме покращенню фінансового становища користувачів та підвищенню їхньої здатності ефективно управляти своїми фінансами.

Економічна доцільність мобільного застосунку Spike також підтверджується можливістю зниження витрат на управління особистими фінансами для користувачів. Використання застосунку дозволить автоматизувати багато рутинних процесів, таких як ведення бюджету, контроль витрат та доходів, планування фінансових цілей тощо. Це сприятиме зниженню витрат часу та ресурсів на управління фінансами, підвищуючи ефективність та точність фінансового планування.

Розробка інформаційного застосунку не завжди повинна бути вигідною, тому необхідно оцінити економічний ефект від її впровадження. Для цього в економічній частині потрібно розрахувати: кошторис капітальних витрат на застосунок, суму загальних інвестиційних витрат та собівартості продукту, вартість реалізації створеного програмного продукту, чистий прибуток, коефіцієнт економічної ефективності.

Головна мета планування процесу розробки – визначення необхідних ресурсів на всіх його етапах, їх рівень залежить від обраного типу застосунку, його складності, особливостей дизайну тощо. Такі етапи наведено у таблиці Б.1.

Таблиця Б.1

Перелік етапів та робіт по розробці застосунку

Найменування		**Вид роботи**		**Виконавець, посада**
стадії	**етапи**	**шифр**	**зміст роботи**	
1	**2**	**3**	**4**	**5**
1 Підготовча стадія	1.1 Вивчення стану питання	1.1.1	Дослідження проблеми	
		1.1.2	Вивчення та аналіз аналогічних розробок	
		1.1.3	Економічне обґрунтування доцільності виконання проекту	
	1.2 Розробка технічного завдання (ТЗ)	1.2.1	Складання плану та розрахунок розробки	

1	2	3	4	5
2 Технічна пропозиція	2.1 Аналіз ТЗ та техніко-економічне обгрунтування проекту	2.1.1	Доведення техніко-економічного обгрунтування	
3 Теоретична розробка	3.1 Теоретичне вивчення задачі	3.1.1	Визначення переліку технологій, які використовуватимуться при розробці та мови програмування	
		3.1.2	Розробка алгоритмів роботи програми на високому рівні	
		3.1.3	Розробка структури програмного забезпечення та схеми взаємодії її компонентів	
4 Практична реалізація	4.1 Розробка дизайну 4.2 Розробка структури сайту	4.1.2	Визначення шаблону дизайну	
		4.2.1	Розробка ядра сайту	
	4.3 Розробка бази даних	4.2.2	Система управління сайтом	
		4.3.1	Визначення системи управління базами даних	
		4.3.2	Розробка структури бази даних	
5. Заключна стадія	5.1 Ознайомлення зацікавлених осіб з проектом	5.1.1	Підготовка презентації	
		5.1.2	Демонстрація системи	
		5.1.3	Навчання роботи з системою	

На основі даного переліку визначається кількість виконавців, тривалість виконання робіт в днях, та рівень оплати праці виконавців по видам робіт, що виконуються. Такі дані приведені у таблиці 5.2.

Таблиця Б.2

Зведені результати тривалості та трудомісткості розробки та реалізації проекту

Шифр роботи	Найменування роботи	Виконавець, посада, спеціальність	Кількість виконавців	Тривалість виконання роботи, дні	Денна тарифна ставка, грн	Оплата праці, грн
1.1.1 - 1.2.1	Дослідження проблеми	Інженер-програміст	1	2	320	640
4.1.1 - 4.3.2	Практична реалізація	Інженер-програміст	1	6	320	1920
5.1.1 - 5.1.3	Заключна стадія	Інженер-програміст	1	1	320	320

До розробки застосунку залучено одного інженера-програміста, зайнятість якого становить 18 днів та керівника практики від підприємства. Розрахунок місячної заробітної плати програміста та керівника виконується згідно формули:

Мз=(Мпз* Крг)*Рд , де

Мпз-погодинна заробітна плата ;

Крг – кількість робочих годин в день (8 год);

Рд – кількість робочих днів;

Мз – заробітна плата в місяць.

Розраховуємо мінімальну місячну зарплату програміста та керівника.

Дані розрахунків заносимо в таблицю:

Таблиця Б.3

Витрати на заробітну плату

Найменування посади	**Місячний посадовий оклад, грн**	**Оплата за робочий день, грн**	**Кількість днів роботи**	**Витрати на заробітну плату, грн**
Керівник	4800	240	9	2160
Інженер-програміст	6400	320	18	5760

Разом: 7920.

Витрати на науково-дослідницьку роботу по розробці програмних засобів та апаратури, щодо даного дипломного проекту, включають наступні елементи:

— витрати на основну заробітну плату виконавців.

Це сума заробітної плати за кожний етап робіт і вона наведена у таблиці Б.2

ЗПосновна= 7920

— додаткова заробітна плата. Її можна обчислити за формулою:

ЗПдодаткова=0,12*ЗПосновна (Б.1)

ЗПдодаткова= 950,4

Таким чином, загальний фонд оплати праці, що обчислюється за формулою:

ФОП=ЗПосновна+ЗПдодаткова (Б.2)

ФОП=8870,4

обов'язкові відрахування на заробітну плату (ЄСВ – єдиний соціальний внесок – 22%). Таким чином обов'язкові відрахування складають:

ЄСВ = ФОП * 0,22 = 1951,5

накладні витрати розраховуємо за формулою:

НВ=0,2*ЗПосновна. (Б.3)

НВ=1584