



УДК 004.9:004.9:336.71

## RESEARCH AND OPTIMIZATION OF TEST DESIGN IN THE DEVELOPMENT OF BANKING SYSTEMS ДОСЛІДЖЕННЯ ТА ОПТИМІЗАЦІЯ ТЕСТ ДИЗАЙНУ ПРИ РОЗРОБЦІ БАНКІВСЬКИХ СИСТЕМ

Samofalova H.M. / Самофалова Г.М.

*QA engineer /інженер з розробки nf тестування програмного забезпечення*

ORCID: 0009-0006-7238-2134

*TestMatick, Cherkasy, Priportova 42/1, 18021**Компанія TestMatick-Україна, Черкаси, вул. Прупортובה, 42/1, 18021*

Kalyta N.I. / Калига Н.І.

*k.t.s., as.prof. /к.т.н., доц.*

ORCID: 0000-0001-6181-732X

*Kharkiv National University of Radioelectronics, Kharkiv, Nauky Ave, 14, 61166**Харківський національний університет радіоелектроніки, Харків, пр. Науки, 14, 61166*

**Анотація.** У статті розглянуто створення оптимального тест плану для тестування електронної банківської системи. Банківські системи є надскладними за архітектурою, вимогами до захисту інформації, інтеграцією з платіжними системами, що обумовлює жорсткі вимоги до якості тестування таких застосунків. На основі аналізу видів, методів та технік тестування розроблено оптимальний тестовий план із атомарних тест-кейсів, проведено його апробацію на прототипі застосунку. Порівняння мануального та автоматизованого тестування дозволяє зробити висновок у доцільності використання стратегії комбінованого тестування розглянутого класу інформаційних систем.

**Ключові слова:** методи тестування, електронні банківські системи, оптимальний тест план, тест кейс.

### Вступ.

Діджиталізація є одним із найважливіших процесів у сучасному світі, адже інформаційні технології знаходяться в активному стані розвитку та впровадження у людське життя. Одним із прикладів діджиталізації є розробка електронних банківських систем (ЕБС). ЕБС забезпечують можливість виконання фінансових операцій за допомогою мобільних застосунків або веб-сайтів, що є суттєво вищим рівнем обслуговування клієнтів без обов'язкової особистої присутності у відділенні банку. Заощаджується час і клієнтів, і персоналу, зокрема стає доступним моніторинг рахунків, проведення різних платежів. Але такі системи є дуже складними у розробці та подальшому їх супроводу, до них висуваються підвищені вимоги щодо надійності зберігання даних, конфіденційності обробки персональних даних, швидкості та захищеності передачі транзакцій тощо.

Тестування є невід'ємним етапом у системному проектуванні та створенні будь-яких інформаційних систем різного рівня складності, а для банківських систем до того ж не є простою задачею, потребує ретельного планування та вимагає від фахівця з тестування певного досвіду і компетентності, оскільки банківські системи мають особливості, які роблять їх надскладними [1, 2]:

– складна архітектура та велика кількість функціональностей, що потребує глибокого аналізу та тестування кожного її елемента;



- необхідність забезпечувати захист персональних даних клієнтів, оскільки будь-які вразливості можуть привести до серйозних наслідків;
- інтеграція з іншими системами, наприклад, з платіжними.

Зважаючи на жорсткі вимоги до ЕБС у різних аспектах і, як наслідок, велику трудомісткість тестування, актуальною є задача створення ефективних тестових планів для перевірки якості застосунків, що виходять на ринок.

Тест-дизайн – це процес створення плану тестування, який включає в себе вибір тестових випадків та встановлення критеріїв їх виконання [3]. Ефективність та якість тестування обумовлена вибором таких тестових кейсів (сценаріїв), які забезпечують тестування всіх функцій та можливостей програми, тобто дають максимальне покриття програмного забезпечення без використання надлишкових тестів.

#### **Мета дослідження.**

Метою дослідження є оптимізація тест-дизайну при розробці електронних банківських систем. В рамках поставленої мети потрібно вирішити наступні задачі: виконати структурно-функціональний аналіз ЕБС, визначити критерії ефективності тестування застосунку, сформулювати задачу оптимізації тест-дизайну, розробити оптимальний тестовий ран із атомарних тест-кейсів, провести його апробацію на прототипі застосунку.

#### **Аналіз сучасних підходів до тестування програмних систем**

Аналіз видів, методів та технік тестування за наперед визначеною класифікацією дозволяє впорядкувати їх та обрати необхідні підходи, що можуть бути використані при розробці тест дизайну для електронної банківської системи. Видів та технік тестування існує досить багато, кожен з них має певні передумови, призначення, і тому запропоновані їх класифікації за різними ознаками [4]: за знанням внутрішньої системи, за об'єктом тестування, за часом та способом його проведення, за критерієм позитивності тощо. Зважаючи на важливість етапу тестування та його ресурсоемність, розроблено достатньо велику кількість інструментальних засобів, які застосовуються для [5]:

- автоматизованого тестування (наприклад, Selenium);
- тестування безпеки;
- тестування навантаження;
- тестування користувачького інтерфейсу;
- контролю якості;
- генерації даних.

В залежності від мети та рівня тестування для досягнення кращих результатів зазвичай не обмежуються одним підходом, а сумісно використовують декілька.

#### **Основний матеріал дослідження.**

Тест план, що покриє тестування всієї електронної банківської системи, можемо визначити на основі аналізу функцій та взаємодії її складових.

Основними складовими електронної банківської системи є [1]:

1. *Інтерфейс користувача.* Інтуїтивно зрозумілий інтерфейс надає можливість користувачам легко взаємодіяти з системою. Клієнтська частина електронної банківської системи включає в себе різноманітні засоби доступу до



банківських послуг, такі як мобільні застосунки, веб-сайти банків, термінали самообслуговування та картки клієнтів.

2. *База даних.* Розробка та підтримка бази даних необхідна для забезпечення та зберігання всієї інформації, пов'язаної з клієнтами, їхніми рахунками, транзакціями та іншою інформацією, що стосується банківської діяльності.

3. *Системи авторизації та ідентифікації.* Розробка та впровадження такої системи забезпечує ідентифікацію та авторизацію користувачів. Для ЕБС вона повинна мати високу надійність та безпеку.

4. *Механізми безпеки.* Ці механізми використовують шифрування та інші методи захисту даних в процесі обміну інформацією між клієнтами та банком. Мають на меті захистити конфіденційну інформацію клієнтів та уникнути шахрайства та крадіжок.

5. *Технічна підтримка.* Вона повинна забезпечувати належну технічну підтримку клієнтів, включаючи допомогу із встановленням та налаштуванням програмного забезпечення, забезпеченням інформації про доступні функції та послуги, а також із допомогою з усуненням технічних помилок і проблем, які виникають під час роботи в ЕБС.

6. *Системи звітності та аналітики.* Це інструменти, які дозволяють збирати, аналізувати та відображати інформацію про фінансові операції та поведінку клієнтів в електронних банківських системах. Результати аналізу таких даних використовуються у прийнятті рішень щодо вдосконалення пропонуванних продуктів та послуг, а також в забезпеченні високого рівня безпеки та захисту від шахрайства.

7. *Комунікаційні мережі.* Це системи, що забезпечують передачу інформації між різними компонентами електронної банківської системи, включаючи сервери, комп'ютери клієнтів та інші пристрої. Ці мережі дозволяють банку підтримувати зв'язок з клієнтами, обмінюватися даними та забезпечувати віддалений доступ до банківських послуг.

Виходячи з (не)функціональних вимог до складових ЕБС, мети та методів тестування, розробляється тестовий дизайн.

Ціль тестування: спочатку необхідно чітко визначити ціль тестування – це може бути перевірка функціональності, продуктивності, безпеки або інших аспектів програмного продукту чи системи.

Вибір методу тестування: залежно від типу програмного продукту можуть використовуватися різні методи тестування – модульне тестування, інтеграційне тестування, системне тестування тощо.

(Не)функціональні вимоги до складових ЕБС: на основі вимог до складових системи створюються тестові сценарії, які мають бути чіткими і повними, щоб покрити різні аспекти програмного продукту і перевірити його відповідність вимогам.

Оптимізація тест-дизайну включає в себе покращення створення тест-кейсів, що потенційно знижує кількість пропущених помилок, підвищує ефективність та забезпечує високу якість продукту.

Таким чином, тестування ЕБС ставить за мету перевірку критично важливих аспектів системи, а саме:



1) функціональність: перевірити правильність виконання функцій застосунку, таких як переказ коштів між рахунками, оплата послуг та товарів, додавання та видалення користувачів тощо;

2) надійність: перевірити стійкість до різних за обсягом та валідністю вхідних даних та сценаріїв використання, зокрема під час одночасного навантаження системи багатьма користувачами;

3) безпека: перевірити, що застосунок забезпечує конфіденційність персональних даних користувачів;

4) відповідність UI/UX вимогам.

Очікуваний результат тестування – визначення помилок та відмов у застосунку для їх подальшого виправлення.

Критерієм ефективності тест дизайну є максимальне покриття критично важливих аспектів системи при обмеженнях на час тестування та кількість QA інженерів, що визначається рамками спринтів проекту (зазвичай подібні проекти розробляються за гнучкою методологією), і задача оптимізації тест дизайну полягає у виборі такої множини тест кейсів, щоб максимально охопити тестуванням програмний застосунок:

$$E = \max_{T_f, T_r, T_s, T_i \in T} G[F(T_f), R(T_r), S(T_s), I(T_i)] \quad (1)$$

при 
$$\sum_{i=1} t_i \leq t_{\max}, \sum_{i=1} p_i \leq p_{\max}, \quad (2)$$

де  $T_f, T_r, T_s, T_i$  - множини тестових кейсів для тестування функціональності, надійності, безпеки та UI/UX відповідно з всієї можливої множини тестів  $T$ , причому кожен тест кейс потребує для виконання певного часу та задіяння певної кількості QA інженерів:

$$T_i = \langle t_i, p_i \rangle. \quad (3)$$

### Апробація тестового дизайну.

Для апробації тестового дизайну необхідно створити банківський застосунок, який складається з трьох частин: форма заповнення, власний кабінет користувача та власний кабінет адміністратора.

Банківська система складається з трьох глобальних елементів: форма заповнення, банківський застосунок для клієнтів, банківський застосунок для співробітників. Бізнес потік системи складається з наступних кроків:

а) клієнт заповнює форму;

б) реєструється в банківському акаунті;

в) відкриває необхідні рахунки;

г) створює кредитні картки;

д) проводить транзакції;

е) адміністратор створює статистичну звітність та проводить моніторинг транзакції за особливих умов.

Банківська система знаходиться на етапі випуску першого білда, що потребує проведення тестування основних аспектів системи для її визначення працездатності та виявлення критичних дефектів для їх подальшого усунення, а саме – димного тестування.



Димне тестування (Smoke test) – це один з основних методів тестування, який зазвичай проводиться перед початком більш широкого тестування. Його основна мета – перевірити, чи є мінімальний прийнятний рівень якості в програмному забезпеченні, щоб відпустити його на більш широкі тести [6]. Воно проводиться зазвичай після кожної нової збірки програмного забезпечення. Цей процес охоплює тестування ключових функціональних можливостей програми, щоб перевірити, чи відповідає вона мінімальним вимогам до якості. Іншими словами, тут відбувається валідація того, що критично важливі частини функціональності системи працюють згідно з вимогами.

Якщо Smoke тестування успішно пройшло, то програмне забезпечення може бути відправлене на більш широке тестування. Якщо ж тестування не пройшло успішно, то розробники повинні виправити помилки, перевірити програмне забезпечення ще раз і повторити Smoke тестування [5].

Поєднання мануального та автоматизованого тестування використовується також для збільшення ефективності пошуку критичних дефектів системи. Мануальне тестування допомагає виявити недоліки, пропущені при автоматизованому тестуванні. Автоматизоване тестування, з іншого боку, може допомогти знизити час і витрати, пов'язані з повторенням однакових тестів. Одночасне проведення мануального та автоматизованого тестування покращує ефективність процесу тестування. Автоматизоване тестування може виконувати тести, які вимагають багато часу та повторення, тоді як мануальне тестування може зосередитися на складних випадках.

Відповідно до обраного підходу тестування необхідно сформулювати наступні тест кейси:

1. Тестування обробки форми заповнення:
  - а) перевірка процесу проходження форми заповнення;
  - б) перевірка валідації полів у формі заповнення.
2. Тестування реєстрації/авторизації користувача:
  - а) перевірка процесу реєстрації клієнта;
  - б) перевірка процесу авторизації клієнта;
  - в) перевірка процесу авторизації адміністратора.
3. Тестування функцій операцій з рахунками:
  - а) перевірка процесу відкриття рахунку;
  - б) перевірка транзакцій між рахунками;
  - в) перевірка історії операцій з рахунком;
4. Тестування дизайну:
  - а) перевірка UI/UX відповідності форми заповнення;
  - б) перевірка UI/UX відповідності акаунту клієнта;
  - в) перевірка UI/UX відповідності акаунту адміністратора.
5. Тестування звітності:
  - а) перевірка можливості генерації звітів про операції між рахунками;
  - б) перевірка можливості генерації звітів про статистичні дані.
6. Тестування коректності відображення даних:
  - а) перевірка коректності відображення даних про рахунки;
  - б) перевірка коректності відображення даних про транзакції;



в) перевірка коректності відображення інформації про дані користувача.

На основі попереднього аналізу методів тестування та визначення його цілей були створені тест кейси для повного покриття критичного функціоналу банківського застосунку (таблиця 1).

**Таблиця 1 – Перелік створених тест кейсів**

Тест кейс	Зміст тест кейсу
ТС11	Перевірка наявності усіх необхідних елементів на форму заповнення, необхідних для повноцінної реєстрації клієнта (поля, підказки, веб сторінки, вибір продукту і т.д.)
ТС12	Перевірка роботи функціоналу форми заповнення у випадку виконання негативних сценаріїв (введення невалідних даних під час проходження)
ТС21	Перевірка функціоналу створення нового акаунта (його конвертації) після проходження форми заповнення
ТС22	Перевірка функціоналу авторизації клієнта у свій кабінет
ТС23	Перевірка функціоналу авторизації адміністратора у свій кабінет
ТС31	Перевірка функціоналу відкриття нових рахунків клієнтом
ТС32	Перевірка функціоналу проведення внутрішніх транзакцій між рахунками клієнта
ТС33	Перевірка правильного відображення історії проведених транзакцій у власному кабінеті клієнта
ТС41	Перевірка відповідності UI дизайну розробленої форми заповнення до дизайну представленого в Figma
ТС42	Перевірка відповідності UI дизайну розробленого персонального кабінету клієнта до дизайну представленого в Figma
ТС43	Перевірка відповідності UI дизайну розробленої персонального кабінету адміністратора до дизайну представленого в Figma
ТС51	Перевірка функціональності створення звітів з проведених транзакцій за певний час
ТС52	Перевірка функціональності створення статистичних звітів
ТС61	Перевірка правильності відображення даних про відкриті рахунки клієнта, які йому доступні
ТС62	Перевірка правильності відображення даних про проведенні транзакції поточним клієнтом, які йому доступні
ТС63	Перевірка правильності відображення даних клієнта, які доступні як тільки для перегляду, так і для редагування

*Авторська розробка*

Результати проведеного мануального тестування наведені у таблиці 2.

Результати проведеного автоматизованого тестування наведені у таблиці 3.



**Таблиця 2 – Результати проведення мануального тестування**

Тест кейс	Результат	Час проведення
ТС11	Пройдено успішно	~ 7 хв
ТС12	Пройдено успішно	~ 5 хв
ТС21	Пройдено успішно	~ 10 хв
ТС22	Пройдено успішно	~ 3 хв
ТС23	Пройдено успішно	~ 3 хв
ТС31	Пройдено успішно	~ 3 хв
ТС32	Пройдено успішно	~ 10 хв
ТС33	Пройдено успішно	~ 10 хв
ТС41	Провалено Причини: 1. Кнопка «Далі» не стає активною при повному заповненні сторінки 2. Обов'язкові для заповнення поля не відмічені знаком «*»	~ 10 хв
ТС42	Провалено Причини: 1. Опції «гамбургер» меню не мають іконок ліворуч 2. Деякі елементи не вирівняні відповідно до Figma	~ 10 хв
ТС43	Провалено Причини: 1. Опції «гамбургер» меню не мають іконок ліворуч 2. Деякі елементи не вирівняні відповідно до Figma	~ 10 хв
ТС51	Пройдено успішно	~ 15 хв
ТС52	Пройдено успішно	~ 15 хв
ТС61	Пройдено успішно	~ 10 хв
ТС62	Пройдено успішно	~ 10 хв
ТС63	Пройдено успішно	~ 10 хв

*Авторська розробка*

**Таблиця 3 – Результати проведення автоматизованого тестування**

Тест кейс	Результат	Час проведення
ТС11	Пройдено успішно	~ 3 хв
ТС12	Пройдено успішно	~ 3 хв
ТС21	Пройдено успішно	~ 5 хв
ТС22	Пройдено успішно	~ 1 хв
ТС23	Пройдено успішно	~ 1 хв
ТС31	Пройдено успішно	~ 1 хв
ТС32	Пройдено успішно	~ 3 хв
ТС33	Пройдено успішно	~ 3 хв
ТС41	не підтримується	—



ТС42	не підтримується	–
ТС43	не підтримується	–
ТС51	Пройдено успішно	~ 5 хв
ТС52	Пройдено успішно	~ 5 хв
ТС61	Провалено Причина: помилка при підключення до БД	~ 5 хв
ТС62	Провалено Причина: помилка при підключення до БД	~ 5 хв
ТС63	Провалено Причина: помилка при підключення до БД	~ 5 хв

*Авторська розробка*

### **Аналіз результатів дослідження.**

Порівняння результатів проведених мануального та автоматизованого тестувань наведено у таблиці 4.

**Таблиця 4 – Порівняння результатів тестування**

Вид тестування	Мануальне	Автоматизоване
Загальний час проведення	~ 2 год 21 хв	~ 45 хв
Середній час проведення тест кейсу	~ 8.3 хв	~ 4.5 хв
Кількість пройдених тест кейсів	17	10
Кількість провалених тест кейсів	3	0
Кількість зіпсованих тест кейсів	0	6 (обрив з'єднання з БД – 3, не підтримується – 3)

*Авторська розробка*

Мануальне тестування та автоматизоване тестування – це дві різні стратегії тестування програмного забезпечення, кожна з яких має свої переваги та недоліки.

Виходячи з отриманих результатів можна зробити висновок, що автоматизоване тестування значно швидше за мануальне, тому що автоматизовані тести можуть бути запуснені без перерв на відпочинок та виконуються на швидкості комп'ютера. Мануальне тестування потребує часу на підготовку, виконання тестів та запису результатів.

Мануальне тестування зазвичай є більш гнучким, оскільки може бути змінене та адаптоване в залежності від потреб та вимог користувача. Автоматизоване тестування може бути менш гнучким, оскільки тести часто створюються для конкретних сценаріїв та вимог. Мануальне тестування може бути більш відповідним вимогам користувача, оскільки тести базуються на досвіді користувача та його потребах. Автоматизовані тести можуть бути створені на основі вимог, але вони можуть бути менш ефективними у випадках, коли вимоги змінюються.





Отже, як бачимо, обидві стратегії тестування мають свої переваги та недоліки, і їх вибір залежить від конкретного проекту та його потреб. Найкраще є використовувати комбінацію обох методів для досягнення найкращого результату тестування. Мануальне тестування може бути використане для тестування нових функцій та ручного тестування користувача, тоді як автоматизоване тестування може бути використане для швидкого виконання багатьох тестів з високою точністю та вартістю.

Також, метод Smoke тестування дозволив знайти критичні проблеми тестованого застосунку, використовуючи мінімальний набір тестових кейсів, оскільки таке тестування спрямоване на швидке виявлення серйозних проблем або критичних помилок, що можуть призвести до недоступності основних функціональних можливостей системи. Це дозволяє забезпечити мінімальний рівень стабільності перед подальшими тестами. Обмежений набір тест-кейсів, спрямованих на перевірку основних функціональностей системи, дозволяє ефективно використовувати ресурси тестувальників, час і зусилля, оскільки вони зосереджуються на найважливіших аспектах програмного забезпечення. Smoke тестування допомагає забезпечити базовий рівень якості, перевіряючи, чи виконуються основні функції та функціональні вимоги системи. Це важливо для того, щоб упевнитися, що система готова до подальшого тестування та використання.

### **Висновки.**

Було розглянуто задачу тестового дизайну та його оптимізації при розробці електронних банківських систем. На основі аналізу функцій таких систем та взаємодії їх складових, аналізу методів тестування були визначені особливості організації та проведення тестування для успішного виходу продукту на ринок.

У тестуванні була задіяна та частина електронної банківської системи, що пройшла стадію розробки, створено профіль клієнта та перевірено функціональні можливості системи і клієнтського інтерфейсу.

Розроблений тест дизайн включає:

- визначення тестового рану для максимального покриття критичного функціоналу системи;
- написання оптимальних та атомарних тестових кейсів за визначеним тестовим раном.

Було проведено Smoke тестування із використанням як мануального, так і автоматизованого підходу, результати якого відображені у тестовому звіті. На практиці результати дослідження можуть застосовуватись в роботі над ІТ проектами по створенню ефективних тестових дизайнів для тестування складних електронних систем, зокрема, електронних банківських систем.

### **Література:**

1. Shelagh Heffernan. Modern Banking in Theory and Practice. Wiley, 1996. 470 с. ISBN 978-0471962090
2. Самофалова Г.М. Дослідження проблем тестування банківських систем //27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь XXI столітті». Зб. матеріалів форуму. Т.6. Конференція «Інформаційні інтелектуальні



системи» – Харків: ХНУРЕ. 2023. – С.125-126.

3. Priyadarshi Tripathy, Kshirasagar Naik. Software Testing and Quality Assurance: Theory and Practice. Wiley-Spektrum, 2015. 648 с. ISBN 978-1-119-09518-7

4. Види тестування та відмінності між ними. URL <https://qagroup.com.ua/publications/vydy-testuvannya-ta-vidminnosti-mizh-nymy/> (дата звернення 10.03.2024).

5. Fairouz Tchier, Ali Mili. Software Testing: Concepts and Operations. Wiley, 2015. 383 с. ISBN 9788126557561.

6. Cem Kaner, Jack Falk, Hung Q. Nguyen. Testing Computer Software. 3-тє вид. Wiley, 1999. 480 с. ISBN 0471358460

***Abstract.** The article discusses the creation of an optimal test plan for testing the electronic banking system. Banking systems are extremely complex in terms of architecture, requirements for information protection, integration with payment systems, which causes strict requirements for the quality of testing such applications. Based on the analysis of types, methods and techniques of testing, an optimal test wound was developed from atomic test cases, and its approval was carried out on the prototype of the application. A comparison of manual and automated testing allows us to conclude that it is appropriate to use a strategy of combined testing of the considered class of information systems.*

***Key words:** testing methods, electronic banking systems, optimal test plan, test case.*

Стаття надіслана: 17.03.2024 г.  
© Самофалова Г.М., Калита Н.І.