

ДОСЛІДЖЕННЯ ПРОБЛЕМ ТЕСТУВАННЯ БАНКІВСЬКИХ СИСТЕМ

Самофалова Г.М.

Науковий керівник – к.т.н., проф. Калита Н.І.

Харківський національний університет радіоелектроніки, каф. СТ
м. Харків, Україна

тел. +380674041639, e-mail: hanna.samofalova@nure.ua

In this work described the problem of organizing the process of testing banking systems. The development of AI QA assistant is proposed as a solution. The assistant will be able to generate a complete test design based on the received system data and the data obtained during its training.

Створення банківських онлайн систем є досить актуальною темою у наш час, оскільки світ активно залучений у процес діджиталізації багатьох сфер життя, і процес виконання фінансових операцій не є виключенням.

Розробка та підтримка банківських систем є досить складним процесом, оскільки вони повинні забезпечувати безпеку транзакцій та взаємодію елементів системи між собою.

Планування, організація та проведення тестування є обов'язковим етапом розробки будь-яких інформаційних систем та застосунків, а для банківських систем є особливо відповідальним, бо кінцевий продукт повинен задовольняти низці вимог [1]:

1. Безпека. Система повинна надавати повну безпеку свої даних та захищати їх від витоку даних або шахрайства.

2. Зручність. Елементи інтерфейсу системи повинні бути максимально інтуїтивно зрозумілі для використання як клієнтів, так і робітників самого банку.

3. Масштабованість. Банківська система повинна витримувати навантаження великої кількості вхідних даних, транзакцій та користувачів.

4. Аналітика. Для удосконалення продукту система повинна збирати та аналізувати дані користувачі та їх поведінку у рамках системи.

Враховуючи повний комплекс вимог, дуже складно організувати тестування, а саме вдосконалення підходів до розробки та застосування тест дизайну. Типовими помилками при підготовці тестування є:

- 1) неправильний вибір інструментів тестування;
- 2) недостатній аналіз вимог та функціональності застосунку, що може призвести до того, що деякі функції не будуть протестовані;
- 3) неправильна оцінка обсягу тестування, що може призвести до того, що застосунок не буде вичерпно протестований.

Навіть такий мінімальний набір проблем може призвести до недостатнього вибору тестових кейсів та тестових сценаріїв, внаслідок чого тестування буде неповним та неякісним.

Задача полягає в розробці такого підходу до організації тестування, щоб можна було максимально підвищити ефективність тестування та знизити ризик виходу неякісного продукту. Пропонується розробка програмного застосунку, який є QA асистентом на базі штучного інтелекту (ШІ), що буде мати доступ до необхідних даних банківської системи. Для навчання такого асистента, повинні бути задіяні як фахівці QA команди, так і розробники системи, що забезпечить опанування QA асистентом повної бази можливих сценаріїв тестування та можливих багів.

Для навчання чат-бота команда тестувальників та розробників може скористатися декількома підходами:

- створити базу тест-кейсів, яка містить різні види тестів та їх описи;
- використовувати техніки генерації тексту на основі навчальних даних для створення тест-кейсів;
- розробити набір правил для генерації тест-кейсів, які відповідають певним вимогам;
- створити базу багів, яка містить різні рівні критичності та їх описи.

Ґрунтуючись на цій базі, QA асистент зможе згенерувати модель тестування. ШІ асистент повинен мати низку корисних функцій, наприклад, аналізу та генерації. Функції аналізу: створення графіків, створення статистики, генерація аналітики на базі зібраних даних. Функції генерації: генерація тестового плану, генерація тестових кейсів за певними критеріями, генерація чеклістів.

При розробці такого QA асистента використовується програмне забезпечення OpenAI, яке призначене для створення чат-ботів. Для формування людиноподібних відповідей асистента використовується механізм "послідовий прогностичний контекст". Це техніка, яка використовується в чат-ботах для покращення їх здатності передбачати наступну дію користувача на основі попередньої історії діалогу. При використанні цього механізму чат-бот запам'ятовує попередні повідомлення користувача і використовує їх, щоб зрозуміти контекст наступного повідомлення та відповісти на нього. Для реалізації цього механізму чат-бот повинен мати здатність запам'ятовувати попередні повідомлення і діалоги. Зазвичай, це досягається за допомогою алгоритмів машинного навчання, які використовують нейронні мережі або моделі на основі правил [2].

Список використаних джерел:

1. Kaner C., Bach J., Pettichord B (2001). Lessons Learned in Software Testing. Wiley.
2. Cem Glenford J. Myers, Corey Sandler, Tom Badgett (2011). The Art of Software Testing. Wiley.