

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»



Факультет комп'ютерних наук та технологій
Кафедра «Комп'ютерні системи та мережі»

ДОРОФЕЄВ СТАНІСЛАВ ОЛЕКСАНДРОВИЧ
Група КНТ-612м

КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВЗАЄМОДІЇ З
ПЛАТФОРМОЮ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ
ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ
АВТОРЕФЕРАТ

магістерської роботи на здобуття освітньо-кваліфікаційного
рівня «магістр» 123 «Комп'ютерна інженерія»
освітньої програми «Спеціалізовані комп'ютерні системи»

2023 р.

Магістерська робота є рукопис.

Робота виконана в Національному університеті «Запорізька політехніка», на кафедрі комп'ютерних систем та мереж

Керівник

кандидат технічних наук, доцент
Ільяшенко Матвій Борисович,
Національний університет «Запорізька
політехніка», доцент кафедри
комп'ютерних систем та мереж

**Офіційний
рецензент:**

Каплієнко Тетяна Ігорівна,
к.т.н., доцент кафедри «Програмних
засобів»

Захист відбудеться "13" грудня 2023р.

Секретар екзаменаційної комісії, доцент кафедри
комп'ютерних систем та мереж **Н.А. Куликовська**

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Актуальність теми дослідження. У процесі розробки IoT-систем, в тому числі IoT-пристроїв, важливим етапом є тестування. Тестування пристроїв Інтернету речей (IoT) є важливим завданням ще на етапі розробки рішення IoT. Тестування дозволяє виявити відповідність між реальною та очікуваною поведінкою пристрою чи програми, провести валідацію, з метою випустити на ринок якісний продукт.

Розробка та впровадження підсистеми, що включає в себе конструктор тестів та моніторинг стану платформи, є багатогранним процесом, спрямованим на підвищення ефективності та надійності автоматизованої платформи. Ця підсистема слугує важливим інтерфейсом між користувачами та автоматизованою платформою, надаючи інструменти для створення тестів та моніторингу робочого стану платформи в режимі реального часу.

На початковому етапі проводиться ретельний збір вимог, щоб визначити конкретні потреби та очікування від автоматизованої платформи. Це закладає основу для подальшої розробки конструктора тестів і компонентів моніторингу стану платформи.

Конструктор тестів задуманий як інструмент, орієнтований на користувача, з інтуїтивно зрозумілим інтерфейсом. Його дизайн зосереджений на простоті, що дозволяє користувачам легко створювати, редагувати та керувати тестовими кейсами. Функціональність для визначення параметрів тесту, вхідних даних та очікуваних результатів дозволяє користувачам формулювати складні сценарії тестування. Для досвідчених користувачів передбачені можливості написання сценаріїв, що дозволяють створювати власні тестові сценарії відповідно до спеціалізованих вимог до тестування. Функції контролю версій реалізовані для відстеження змін у тестових кейсах, забезпечуючи прозорість і відстежуваність протягом усього процесу тестування.

Водночас, компонент моніторингу стану платформи розроблений таким чином, щоб надавати інформацію про стан автоматизованої платформи в реальному часі. Це передбачає постійне відстеження стану обладнання, програмних компонентів і загальної продуктивності системи. Інтегрована система сповіщення

для оперативного інформування відповідних зацікавлених сторін про будь-які порушення, виявлені в процесі моніторингу. Створено надійну систему реєстрації та звітності для ведення всебічного обліку робочого стану платформи з плином часу, що полегшує пост-аналіз та усунення несправностей.

Мета і завдання дослідження. Метою дослідження є розробка та реалізація підсистеми, що складається з конструктора тестів та моніторингу стану платформи, для взаємодії з автоматизованою платформою. Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Аналіз існуючих рішень у галузі інформаційних систем у форматі конструкторів;

2. Вибір оптимальних засобів розробки з урахуванням виявлених критеріїв;

3. Розробка, тестування WEB-додатку по взаємодії з платформою автоматизованого тестування пристроїв IoT.

Об'єктом дослідження – є клас підсистем взаємодії з платформ тестування.

Предмет дослідження – підсистема з конструктора тестів моніторинг стану в платформі тестування пристроїв Інтернету речей.

Методи дослідження - аналіз, порівняння, систематизація та узагальнення даних про існуючі та розроблені способи автоматизації тестування та створення конструкторів; апробація сучасних WEB-технологій під час побудови веб-додатку, тестування роботи інтерфейсу на ПК, аналіз технологій та підходів, що дозволяють масштабувати та оптимізувати WEB-додаток;

Наукова новизна отриманих результатів:

Робота відноситься до галузі досліджень застосування інформаційних та WEB-технологій для створення інтерфейсу до ядра тестування пристроїв IoT.

Практичне значення отриманих результатів:

Виявлено, обгрунтовано та описано переваги інформаційних технологій як інструменту тестування. Показано, що за допомогою цього інструменту стає можливим підвищити ефективність та зручність процесу тестування нових пристроїв.

Апробація результатів магістерської роботи. Основні положення магістерської роботи та результати досліджень подано до участі на конференції:

- Заходи безпеки для кінцевих пристроїв інтернету речей у системі туманних обчислень / С. О. Дорофєєв [М.Б. Ілляшенко, Н.А. Куликовська, А.В. Тіменко] // Інформаційні технології: економіка, техніка, освіта '2023 : 36. матеріалів XIV Міжнар. науково-практ. конф. молодих вчен., Київ, 26–27 жовт. 2023 р. – Київ, 2023. – С. 133–134.

Структура та обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, додатку. Основна частина містить 61 сторінок, 31 рисунок, список використаних джерел зі 15 найменування.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У першому розділі проведено аналіз сучасних інтелектуальних систем керування рухом. Інтелектуальні системи регулювання дорожнього руху мають значний потенціал для вирішення проблем заторів, підвищення пропускну здатності та безпеки на дорогах. Вони можуть аналізувати дані в реальному часі, прогнозувати умови руху, оптимізувати роботу світлофорів. Однак на практиці впровадження таких систем нашкодується на складнощі.

У результаті аналізу було розглянуто 3 конструктори додатків:

Fliabit - візуальний конструктор додатків на базі технології Qt і мови QML. Дає змогу швидко створювати прототипи і невеликі додатки за допомогою візуального інтерфейсу і бібліотеки віджетів.

Appy Pie - конструктор мультиплатформових додатків на базі HTML5. Надає зручний інтерфейс drag-and-drop та інтеграцію з різними сервісами. Дає змогу публікувати додатки в AppStore і Google Play.

Appery.io - ще один конструктор додатків з інтерфейсом drag-and-drop. Дозволяє створювати нативні, веб і гібридні застосунки. Має безліч додаткових функцій і підтримку останніх версій iOS і веб-технологій.

За результатами аналізу для розробки конструктора тестів обрано підхід на базі HTML5, як найбільш мультиплатформний. Будуть використані схожі рішення з простим інтерфейсом drag-and-drop.

У другому розділі розроблена інтелектуальна система регулювання руху на перехрестях призначена для адаптивного управління світлофорами залежно від реальних умов трафіку. Вона використовує дані віде Було проведено порівняння архітектурних відмінностей, знайшли сильні та слабкі сторони кожного фреймворку та порівняли сфери застосування. Angular, React та Vue знаходяться у стадії дуже активної розробки. Вони регулярно випускають нові версії та підтримують існуючі. Оскільки поточний рівень підтримки в кожному випадку високий, можна без побоювань використовувати будь-який з цих кадрів.

Важливо відзначити, що Angular росте не так швидко, як раніше, тоді як Vue, хоч і з'явився відносно недавно, сильно зростає. Важко передбачити, які фреймворки залишаться актуальними у довгостроковій перспективі, але за кожним із них стоїть велика спільнота, і вони постійно розвиваються.

Результатом дослідження можна позначити наступне: виходячи з вимог до складності використання та масштабованості, а також з огляду на відсутність досвіду в будь-якому з цих фреймворків, було обрано варіант з реалізацією на React.

У третьому розділі описується розробка підсистеми взаємодії для платформи автоматизованого тестування пристроїв Інтернету речей.

Платформа логічно розділена на 3 частини: frontend, backend і ядро тестування. Використовуються бази даних PostgreSQL і Redis.

Розроблювана підсистема взаємодії містить конструктор тестів і моніторинг стану платформи.

Конструктор тестів дає змогу візуально створювати сценарії тестування з базових сутностей "команда", "тест" і "сценарій". Реалізовано сторінки перегляду та створення цих сутностей.

Моніторинг стану використовує дані з Redis для відображення поточного статусу платформи, списку пристроїв, що тестуються, і результатів тестування. Також реалізовано бот у Telegram для надсилання повідомлень про завершення чергового тестування і запиту поточного стану платформи.

Таким чином, розроблена підсистема спрощує створення тестів і дає змогу відстежувати поточний стан процесу тестування.

ВИСНОВКИ

В результаті виконання випускної кваліфікаційної роботи було проведено дослідження конструкторів, порівняльний аналіз веб-технологій Angular, React та Vue, розроблена підсистема взаємодії платформа автоматизованого тестування пристроїв Інтернету речей.

Дослідженні конструкторів було розглянуто особливості сучасних конструкторів, відзначено їх ключові недоліки, що роблять їх використання для вирішення завдання тестування недоцільним.

Аналізі веб-технологій було здійснено повноцінне порівняння трьох популярних сучасних технологій. У порівнянні враховувалася популярність, задоволеність та інші метрики. Також було оцінено продуктивність. Як технологія, що використовується в клієнтській програмі, був обраний найбільш відповідний за вимогами React.

Підсистема взаємодії з платформою включає клієнтський додаток на React, серверний додаток, що працює по API для обміну даними між клієнтським додатком і БД, а також БД PostgreSQL і Redis, що містять інформацію, що використовується в веб-додатку.

Підсистема як частина загального рішення дозволила додатково автоматизувати та спростити роботу тестувальників, а також зробити процес виявлення неполадок ефективнішим. Автоматизація дозволила значно скоротити час випробувань, порівняно з ручним тестуванням.