

УДК 004.89:616.1

DOI: 10.31673/2412-9070.2023.038090

О. І. БАНДУРКА, доктор філософії;

О. В. СВИНЧУК, канд. фіз.-мат. наук, доцент;

О. А. ДАЦЮК, ст. викладач,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ ЯКОСТІ МЕДИЧНИХ ВЕБЗАСТОСУНКІВ У ХМАРНОМУ СЕРЕДОВИЩІ

Розглянуто основні переваги автоматизованого тестування, а також випадки, коли його використання є незалишним. Також досліджено життєвий цикл тестування програмного продукту. Тестування програмного забезпечення проводиться для виявлення дефектів та підвищення якості продукту і є процесом аналізу програмного засобу та документації, що його супроводжує. Тестування є одним із найважливіших та ключових етапів розроблення програмного продукту, особливо, якщо застосунок належить до медичної сфери застосування і пов'язаний зі здоров'ям людини. Метою роботи є реалізація автоматизованого тестування кардіологічних вебзастосунків у хмарному середовищі. Реалізоване автоматизоване тестування має бути кросбраузерним та створювати тест-репорт для підбиття підсумків тестових заходів і результатів. У процесі реалізації автоматизованого тестування мають бути задокументовані тестові артефакти, зокрема тест-план та тест-кейси.

Ключові слова: автоматизоване тестування; вебзастосунок; тестові артефакти; серцево-судинні захворювання.

Вступ

У сучасному світі доволі гостро постає проблема збільшення кількості серцево-судинних захворювань серед населення різних вікових груп. Для розуміння масштабності проблеми зазначимо, що в Україні 30% чоловіків та 15% жінок не доживають до пенсійного віку [1]. Серцево-судинні захворювання (ССЗ) відіграють особливу роль у цій проблемі, а Україна, на жаль, продовжує посідати найвищі позиції за смертністю через цей тип захворювань. Отже, наразі спостерігається надзвичайно велика прірва між станом здоров'я середньостатистичного громадянина Західної (зокрема й України) та Східної Європи. І якщо, наприклад, у 1991 році показник захворюваності ССЗ в Україні на 30% перевищував відповідний середньоєвропейський рівень, то за останні роки рівень захворюваності в країнах Східної Європи вже вдвічі менший порівняно з Україною [2]. Саме тому спостереження за здоров'ям серцево-судинної системи пацієнтів є надзвичайно важливим у нашій країні.

Сьогодні значного поширення набувають різноманітні медичні застосунки. Це пов'язано з тим, що вони зберігають час лікарів та пацієнтів; пацієнт має вільний доступ до перегляду результатів досліджень та аналізів, історії хвороби та внесення необхідних даних, які стосуються стану здоров'я; більш точно можна визначити діагноз, його особливості, отримати діаграми, аналізувати вибрані показники стану, сформувати таблиці та способи лікування і підтримання чудового самопочуття пацієнта — і це все з будь-якої точки планети, бо немає потреби запам'ятовувати великі обсяги інформації та мати при собі дані про пацієнтів у друкованому вигляді. Саме тому такі застосунки необхідно ретельно тестувати, щоб мати точні введені дані, результати аналізу та адекватну роботу програмного продукту. Тестування потрібно ще й через те, що люди постійно припускаються помилок, причому деякі з них можуть бути незначними, тоді як інші — мати руйнівні наслідки. Усе, що виробляється людиною, може містити помилки. Саме тому будь-який продукт потребує перевірки — тестування, перш ніж його можна буде ефективно і безпечно використовувати.

Аналіз літературних даних та постановка проблеми. Пацієнти із серцево-судинними захворюваннями завжди є вразливою групою людей, а пацієнти з інфарктом міокарда мають підвищений ризик захворюваності та смертності [3]. Доволі часто лікарі навіть не мають часового ресурсу щоденно відслідковувати показники пацієнта, а пацієнти із серцево-судинними захворюваннями не завжди здатні вчасно прийти на консультацію до лікаря. Тому наявні кардіологічні вебзастосунки, наприклад RMCARDIO, EKGDX, HeartScan, Cardiolyse, Kardia, Cardiospy Mobile ECG тощо, завдяки щоденному моніторингу дають можливість лікарям відстежувати стан діяльності серця пацієнта та запобігати розвитку важких ускладнень зі зменшенням летальних випадків. У застосунку лікар може відслідковувати електрокардіограму пацієнта у першому типі біполярного відведення кінцівок (рис. 1) та кількість ударів серця за хвилину [4].

У кардіологічному вебзастосунку також відображено середній QRS-комплекс, який допомагає лікарю не переглядати всю довгу стрічку електрокардіограми, оскільки програмний продукт спочатку обчислює амплітуду зубців P, Q, R, S, T, далі виводить їх середнє, а потім малює один комплекс, з огляду на середні дані [5]. Усі ці дані є надто важливими показниками загального стану пацієнта із серцево-судинним захворюванням, але також дуже важливо, щоб на ці дані коректно зв'язали з подальшим відображенням.

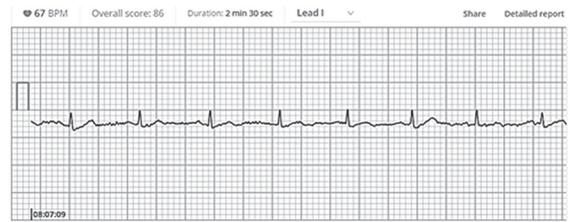


Рис. 1. Зразок ЕКГ пацієнта

Якщо портативний пристрій для ЕКГ зчитав дані правильно, але застосунок намалював кардіограму чи порахував амплітуду неправильно, то у здорової людини можуть виявити ішемію міокарда чи аритмію, а у хворої в потрібний час не помітити ознаки інфаркту міокарда. Тому створена система автоматизованого тестування має охоплювати всі ділянки кардіологічного застосунку та відслідковувати його поведінку, а в разі неочікуваних результатів сповіщати про це розробників.

Автоматизація розглядуваного процесу є актуальним завданням сьогодення, оскільки під час ручного тестування інженер усі тестові справи виконує вручну, а відтак, витрачається багато часу, щоб підготувати набір даних і сценарій та запустити правильні входи чи дії для тестування описаного сценарію [6].

Метою дослідження є моніторинг працездатності кардіологічного вебзастосунку в хмарному середовищі. Реалізація автоматизованого тестування програмного продукту дасть можливість заощадити час для мануального тестування продукту та зменшить ризик припуститися помилок, які можуть призвести до неправильної роботи застосунку та отримання хибних результатів.

Основна частина

Підготовка до автоматизації тестування кардіологічного вебзастосунку передбачає такі етапи (рис. 2):

- аналіз вимог із наведенням уточнювальних запитань щодо застосунку та визначенням обсягу роботи;
- тестове планування зі встановленням приблизного оцінювання часу на тестування, вибором тестових інструментів, що найбільш відповідають архітектурі застосунку;
- тестовий дизайн із підготовки вхідних даних для тестування, розроблення тестової архітектури застосунку;
- тестове виконання з програмуванням створених на попередніх етапах тест-кейсів, баг-репортів;
- тестове завершення зі складанням тест-репорту, де описано кількість багів, їх відсоткове співвідношення, в яких частинах продукту їх траплялося найбільше.

Структуру програми можна подати багатьма модулями, головні з яких такі:

- модуль URL надає утиліти для вирішення та розбору URL-адрес, тобто розбиває вебадресу на зручні для читання частини;
- модуль Chai має кілька інтерфейсів, які дають можливість розробнику вибрати найбільш комфортний. BDD (Behavior Driven Development) підтримує та надає виразний стиль мови та чіткість (читабельність), тоді як стиль TDD (Test Driven Development) забезпечує більш класичний вигляд. Chai доступний як для node.js, так і для браузера і сумісний із будь-яким потрібним тестовим фреймворком. API є досить гнучким, що будь-які синхронні завдання можуть бути легко інкапсульовані в межах одного твердження та повторно використані протягом тестів;
- модуль Nightmare — це високорівнева браузерна автоматизована бібліотека, яка допомагає розкрити кілька простих методів, що імітують дії користувача (наприклад, goto, type та click), з API, який бачить синхронність для кожного блока сценарію;
- модуль Open активно підтримується та виправляє більшість відкритих вихідних проблем, також він підтримує WSL-шляхи до програм Windows.

Для проведення тестування застосунку було використано наведені далі тестові артефакти [8; 9].

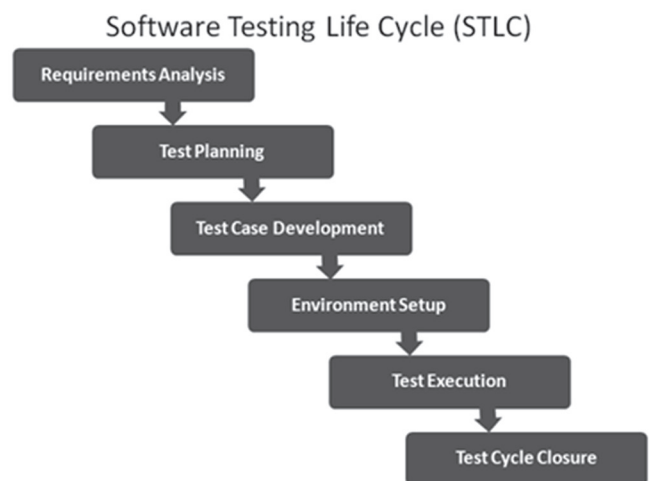


Рис. 2. Життєвий цикл тестування програмного продукту [7]

1. План тестування (Test Plan) — це документ для опису обсягу, ресурсу, підходу до тестування та його графік. Він містить опис тестованих об'єктів, стратегію, розклад, критерій початку і закінчення тестування, потрібного в процесі роботи обладнання, спеціальні знання, а також оцінювання ризиків із варіантами їх розв'язання. Також у тест-плані можуть бути описані ризики, що потребують планування дій у надзвичайних ситуаціях.

2. Набір тест-кейсів і тестів (Test Case & Test suite) — це послідовність дій, за якою можна перевірити чи відповідає тестована функція встановленим вимогам.

3. Дефекти/баг-репорт (Bug Reports/Defects) — це документи, що описують ситуацію або послідовність дій, які призводять до некоректної роботи об'єкта тестування, із зазначенням причин і очікуваного результату.

Для реалізації автоматизованого тестування кардіологічного вебзастосунок було розроблено систему автотестів, яку було розбито на тестові набори, кожен із яких відповідав окремому функціоналу програми: авторизації, реєстраційній формі, реєстрації ролі, інформаційній сторінці пацієнта (таблиця).

Тест-кейси на функціонал інформаційної сторінки пацієнта в кардіологічному вебзастосунку

Тестований функціонал	Очікувані результати
Користувач натискає на кнопку з ім'ям пацієнта зі списку всіх пацієнтів	З'являється сторінка з блоком загальної інформації про пацієнта та блоком інформації зі щоденної електрокардіограми пацієнта
Користувач натискає на кнопку з ім'ям пацієнта зі списку всіх пацієнтів	У блоці міститься інформація щодо BPM (кількість ударів серця за хвилину), а також загальний показник стану здоров'я, тривалість знімання електрокардіограми, кнопка вибору біполярного відведення кінцівок, кнопка для змоги поділитися в застосунку з іншими лікарями інформацією щодо пацієнта, кнопка показу детального репорту, електрокардіограма, середній комплекс QRS, блок з ЕКГ-параметрами, блок кодів Міннесоти (стандартів та класифікації захворювань), блок амплітуди кожного зубця електрокардіограми, блок синдромних кодів (стандартів та класифікації захворювань), блок параметрів варіабельності серцевого ритму, комплексне резюме щоденного стану пацієнта
Користувач натискає на кнопку вибору біполярного відведення кінцівок	З'являється поп-ап вибору біполярного відведення кінцівок із трьох типів
Користувач натискає на будь-який тип біполярного відведення кінцівок	Електрокардіограма, середній комплекс QRS та блок амплітуди кожного зубця електрокардіограми відображається залежно до типу біполярного відведення кінцівок
Користувач прокручує сторінку до блока з ЕКГ-параметрами	Блок ЕКГ-параметрів містить інтервал QT (тривалість деполяризації шлуночків та реполяризації), інтервал коригованого QTc, інтервал коригованого QTcF (за формулою лікаря Фрідерісії), інтервал RR: тривалість серцевого циклу шлуночків (показник частоти шлуночків), тривалість QRS: тривалість деполяризації м'язів шлуночків, вісь QRS, яка є головним вектором активації шлуночків, котрий є загальним напрямком електричної активності
Користувач прокручує сторінку до блока амплітуди кожного зубця електрокардіограми	Блок амплітуди містить значення типу біполярного відведення кінцівок та таких зубців: P, Q, R, S, T, J, J40, J80
Користувач прокручує сторінку до блока параметрів варіабельності серцевого ритму та натискає на стрілочку праворуч	З'являється блок із діаграмою інтервалу між двома серцебиттями (R зубця в комплексі QRS/ЕКГ)
Користувач натискає на стрілочку праворуч у блоці з діаграмою інтервалу між двома серцебиттями	З'являється блок із такими частотними параметрами: VLF (дуже низька частота), LF (низька частота), HF (висока частота)
Користувач натискає на стрілочку ліворуч у блоці з діаграмою інтервалу між двома серцебиттями	З'являється блок параметрів варіабельності серцевого ритму
Користувач натискає на стрілочку ліворуч у блоці з частотними параметрами	З'являється блок із діаграмою інтервалу між двома серцебиттями (R зубця в комплексі QRS/ЕКГ)
Користувач наводить курсором на блок Overall wellbeing Info (загальне самопочуття)	З'являється поп-ап Overall wellbeing Info (загальне самопочуття) з описом стадій самопочуття залежно від відсотка

Закінчення таблиці

Тестований функціонал	Очікувані результати
Користувач наводить курсором на блок Myocardial score (оцінювання міокарда)	З'являється поп-ап Myocardial score (оцінювання міокарда) з описом стану міокарда залежно від відсотка
Користувач натискає на кнопку «скачати PDF»	З'являється вікно друку із сформованим звітом щоденного стану пацієнта
Користувач натискає на кнопку «скачати Excel»	Файл Excel автоматично формується та завантажується до ПК
Користувач натискає на кнопку «скачати RAW»	Файл із розширенням .edf автоматично формується та завантажується до ПК
Користувач натискає на запис зі списку щоденних електрокардіограм.	Блок щоденної інформації пацієнта змінює інформацію згідно з датою, яку вибрав користувач.

Після проходження всіх тестів кардіологічного вебзастосування користувачеві відображається тест-репорт на основі тестових наборів у вигляді вебсторінки (рис. 3), а саме: діаграма часу проходження кожного набору до загальної кількості часу, графік часу проходження, кількість непройдених тестів за кожним набором, детальний опис непройдених тестів, загальна кількість тестів, діаграма часу проходження тестів у одному наборі та багато іншої потрібної інформації.

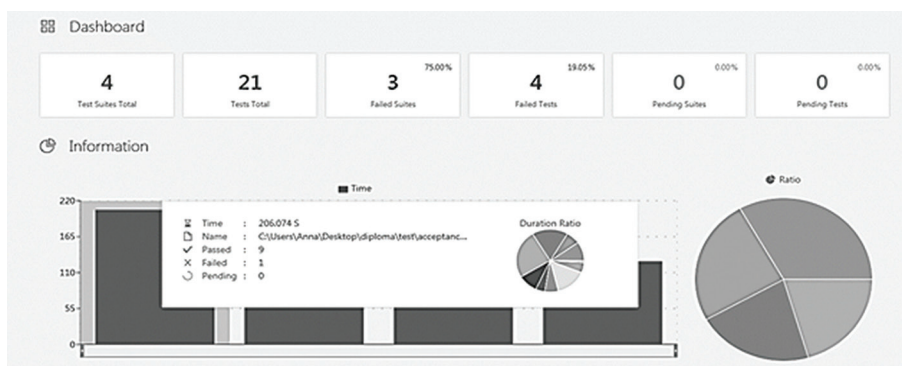


Рис. 3. Сформований тест-репорт до тестування кардіологічного вебзастосування

На верхній панелі виводяться: кількість тест-наборів; загальна кількість тестів; кількість наборів, в яких було виявлено помилки; загальна кількість помилок; кількість наборів, в яких тести не дочекались відповіді від сервера кардіологічного вебзастосування за встановлений час; загальна кількість тестів, які не дочекались відповіді від сервера кардіологічного вебзастосування за встановлений час.

Також у сформованому тест-репорті користувач може відслідковувати час проходження кожного тест-набору у вигляді графіка, в якому шкалу поділено на секунди (рис. 4). Стовпці розташовуються зліва направо відповідно до порядкового номера їх проходження, а за їх висотою визначається час проходження кожного тест-набору. Якщо навести на стовпець бажаного тест-набору, то можна дізнатися, скільки часу забирав кожен тест у співвідношенні до всього набору тестів з інформацією про точний час проходження набору, ім'я набору, кількість пройдених у ньому тестів, кількість помилок, кількість тестів, які не дочекались відповіді від сервера кардіологічного вебзастосування за встановлений час та діаграму співвідношення часу.

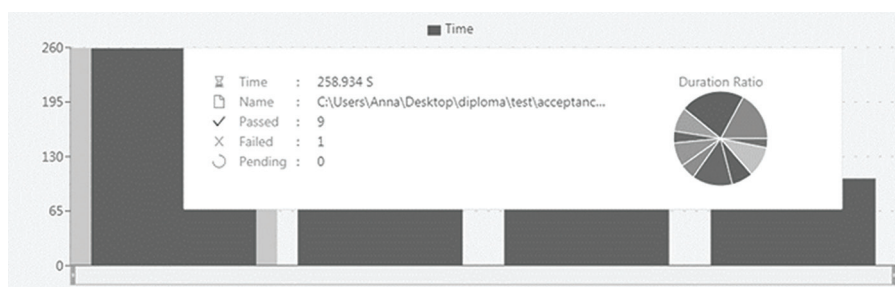


Рис. 4. Часова діаграма тестування кожного кейса з тест-набору «Рестрація»

Під блоком діаграми та графіка користувач може продивитися більш детальну інформацію щодо проходження тест-наборів (рис. 5), а саме: назву набору, час його проходження, статус проходження тестів у ньому, наслідок. Сірим кольором позначено набори, в яких тести виявили помилки у функціоналі кардіологічного вебзастосунку.

File	UseTime	Status	Action
C:\Users\Anna\Desktop\diploma\test\acceptance\signin.test.js	02:33.096	All Passed 4	✓
C:\Users\Anna\Desktop\diploma\test\acceptance\signUp.test.js	03:05.883	4 2	Info
C:\Users\Anna\Desktop\diploma\test\acceptance\registrationForm.test.js	04:18.934	9 1	Info
C:\Users\Anna\Desktop\diploma\test\acceptance\login.test.js	01:43.729	All Passed 1	✓

Рис. 5. Блок «Деталі»

Систему автоматизованих тестів можна використовувати для тривалої повторюваної перевірки вебзастосунку та проводити навантажувальне тестування, що допоможе вирішити питання оптимізації застосунку. Створена система допоможе розробникам кардіологічного вебзастосунку швидко та ефективно реагувати на помилки, менше витратити час на пошук документальної інформації про некоректні результати роботи ресурсу.

Висновки

У результаті дослідження було створено систему автоматизованого тестування кардіологічного вебзастосунку для тривалої повторюваної перевірки частин програми, коли потрібно уникнути «людського фактора». Ця система допоможе здійснювати набагато швидше тестування порівняно з мануальним. Завдяки автоматизованому тестуванню вебзастосунки матиме змогу проводити навантажувальне тестування, що сприятиме вирішенню питання оптимізації застосунку. Подальшим перспективним розвитком дослідження є створення системи гарантування якості медичних застосунків у розподіленому середовищі.

Список використаної літератури

1. *Серцево-судинні захворювання — головна причина смерті українців [Електронний ресурс]. URL: <https://phc.org.ua/news/sercevo-sudinni-zakhvoryuvannya-golovna-prichina-smerti-ukrainciv-visnovki-z-doslidzhennya> (дата звернення 20.08.2023)*
2. *Кардіологічні патології і COVID-19: як зберегти життя пацієнта? [Електронний ресурс]. URL: <http://amnu.gov.ua/kardiologichni-patologiyi-i-covid-19-yak-zberegty-zhyttya-paciyenta/> (дата звернення 20.08.2023)*
3. *Deaths from cardiac arrests have surged in New York City [Електронний ресурс]. URL: <https://www.economist.com/graphic-detail/2020/04/13/deaths-from-cardiac-arrests-have-surged-in-new-york-city> (дата звернення 22.08.2023)*
4. *Як запобігти серцево-судинним захворюванням [Електронний ресурс]. URL: <https://intent.press/expert/health/2023/yak-zapobigti-sercevo-sudinnim-zahvoryuvannjam/> (дата звернення 25.08.2023)*
5. *Understanding LSTM Networks [Електронний ресурс]. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (дата звернення 25.08.2023)*
6. *Knott D. Hands-On Mobile App Testing. Crawfordsville: Addison-Wesley Professional, 2015. 254 p.*
7. *Сапон О. М., Бандурка О. І. Система гарантування якості медичних додатків в розподіленому середовищі // Матеріали XVIII міжнародної науково-практичної конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики». Київ, 23-25 квітня 2020 р. С. 112–113.*
8. *Gundecha U., Cocchiario C. Learn Selenium: Build data-driven test frameworks for mobile and web applications with Selenium Web Driver 3. 2019. 536 p.*
9. *Verma N. Mobile Test Automation with Appium: Mobile application testing made easy. Nishant Verma. 2017. 256 p.*
10. *Тулук А. С., Бандурка О. І. Автоматизоване тестування кардіологічних веб-додатків в хмарному середовищі // Матеріали XVIII міжнародної науково-практичної конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики». Київ, 23-25 квітня 2020 р. С. 78–79.*

O. I. Bandurka, O. V. Svyinchuk, O. A. Datsiuk

AUTOMATED QUALITY TESTING OF MEDICAL WEB APPLICATIONS IN THE CLOUD ENVIRONMENT

In the modern world, the problem of increasing the number of cardiovascular diseases among the population of different age groups is particularly acute. Therefore, monitoring the health of the cardiovascular system of patients is extremely important in our country. Existing cardiology web applications, thanks to daily monitoring, enable doctors to monitor the condition of the patient's heart and prevent the development of severe complications with a decrease in deaths. If the portable EKG device read the data correctly, but the application drew a cardiogram, then a healthy person may detect myocardial ischemia or arrhythmia, and a patient may not notice the signs of a myocardial infarction at the right time. Testing is one of the most important and key stages of software product development, especially if the application refers to the medical field of application and is related to human health. Therefore, the automation of this process is an urgent task today, because during manual testing, the engineer performs all the test cases manually and a lot of time is spent to prepare the data set and the scenario and run the correct inputs or actions to test the described scenario.

The main advantages of automated testing, as well as cases when its use is indispensable, are considered. The life cycle of software product testing is also considered. Software testing is conducted to identify defects and improve the quality of the product and is a process of analyzing the software and the accompanying documentation that accompanies it. The purpose of the work is the implementation of automated testing of cardiology web applications in the cloud environment. Implemented automated testing should be cross-browser and generate a test report to summarize test activities and results. During the implementation of automated testing, test artifacts such as the test plan and test cases must be documented.

Keywords: automated testing; web application; test artifacts; cardiovascular diseases.

