

ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА ДИАГНОСТИКИ И КОРРЕКЦИИ ПОСТИНСУЛЬТНОГО БОЛЕВОГО СИНДРОМА

Шумская Я.П., Кушнер А.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Косарева Е.М. – магистр, ассистент кафедры ПИКС

Аннотация. Экспериментально исследовано функционирование программного средства диагностики и коррекции постинсультного болевого синдрома, предназначенного для автоматизации процессов ведения пациентов, проведения медицинских тестов (VAS, DN4, CSI, HADS), а также формирования диагностических заключений и рекомендаций по лечению. Было выполнено более 60 тест-кейсов, охватывающих ключевые пользовательские сценарии, включая регистрацию и аутентификацию пользователей, управление пациентами, прохождение медицинских тестов и разграничение прав доступа. Успешно пройденных тест-кейсов – 75%. Установлено, что наибольшее влияние на корректность функционирования системы оказывают дефекты, приводящие к созданию некорректных данных и нарушению последовательности выполнения обязательных сценариев тестирования. Предложены рекомендации по улучшению программного средства, включающие усиление серверной валидации, доработку бизнес-логики, унификацию пользовательского интерфейса и т.д.

Ключевые слова: медицина, постинсультный болевой синдром, ручное тестирование, тест-кейсы, бизнес-логика, REST-архитектура, автоматизация медицины

Введение. Сфера медицины все стремительнее развивается с каждым годом. Она опирается не только на привычные исследования и методы работы с пациентами, но и на различные программные решения. Последние помогают работникам медицинской сферы принимать решения, уточнять диагнозы и многое другое. Именно поэтому к подобным программным средствам предъявляются серьезные требования к качеству и безопасности. В таком тестировании важно учитывать даже маленькие дефекты, поскольку они могут привести к серьезным последствиям.

В данной работе рассматривается процесс ручного тестирования веб-приложения, направленного на автоматизацию процесса прохождения тестов на диагностику пациентов с болью после инсульта и предложения врачу рекомендованного лечения, а также автоматизацию формирования заключений. Основным методом тестирования – ручное функциональное тестирование по формализованным тест-кейсам. Ручное тестирование позволяет выявить не только критические ошибки бизнес-логики, но и недостатки пользовательского интерфейса.

Основная часть. Тестируемое программное средство представляет собой веб-приложение для врачей и пациентов, предназначенное для автоматизации процесса диагностики и отслеживания истории лечения и пройденных медицинских тестов. Система построена по клиент-серверной архитектуре с использованием REST-взаимодействия. Язык программирования – Python. Для врачей приложение предлагает автоматизацию ведения базы пациентов с их схемой лечения, список доступных лекарств, а также четыре вида медицинских тестов для помощи в постановке диагноза. Приложение также позволяет отслеживать историю диагнозов и проведенных тестов, корректировать методы лечения. Для пациентов доступен функционал отслеживания истории медицинских тестов, просмотр и отслеживание диагноза и схемы лечения, включая назначенные препараты.

Функциональное тестирование проводилось вручную по заранее подготовленным тест-кейсам, включая как позитивные, так и негативные сценарии. Для каждого теста были зафиксированы предусловия, последовательность шагов, ожидаемый и фактический результаты выполнения.

Ручное тестирование является процессом, в ходе которого QA-специалисты проверяют функциональность программного обеспечения, имитируя действия пользователя [1]. Это делается с целью выявления дефектов и ошибок, которые могут повлиять на его работу.

В процессе тестирования было разработано и выполнено 60 тест-кейсов. Они охватывают процессы регистрации, авторизации, управления пациентами и их карточками, медицинские тесты и разграничение прав доступа. Фрагмент разработанных текст-кейсов приведен в таблице 1.

Таблица 1 – Фрагмент разработанных тест-кейсов

ID	Название теста	Предусловие	Шаги	Ожидаемый результат
РЕГИСТРАЦИЯ (ШАГ 1 – ЛИЧНЫЕ ДАННЫЕ)				
TC-REG-001	Успешная регистрация врача – корректные данные на Шаг 1	Открыта страница /accounts/register/. Пользователь не авторизован.	1. Ввести корректное Имя 2. Ввести корректное Отчество 3. Ввести корректную Фамилию 4. Ввести валидный Email 5. Ввести пароль ≥ 1 спец. символ, ≥ 1 цифра, ≥ 1 заглавная 6. Нажать «Продолжить»	Редирект 302 на /accounts/register + 200 с эндпоинтом /accounts/register/professional/.
TC-REG-002	Поле «Имя»: цифры запрещены	Страница регистрации Шаг 1 открыта.	1. Ввести «Иван1» в поле Имя 2. Нажать «Продолжить»	Красная рамка поля, текст «Цифры и спец. символы запрещены». Переход заблокирован.
TC-REG-003	Поле «Имя»: спецсимволы запрещены	Страница регистрации Шаг 1 открыта.	1. Ввести «Иван@» в поле Имя 2. Нажать «Продолжить»	Красная рамка + «Цифры и спец. символы запрещены». Переход заблокирован.
TC-REG-004	Поле «Имя»: пустое значение	Страница регистрации Шаг 1 открыта.	1. Оставить поле Имя пустым 2. Нажать «Продолжить»	Красная рамка + «Вы не ввели имя». Переход заблокирован.
TC-REG-005	Поле «Email»: пустое значение	Все поля кроме Email заполнены корректно.	1. Оставить поле Email пустым 2. Нажать «Продолжить»	Красная рамка + «Вы не ввели свой email». Переход заблокирован.

По результатам верификации 45 тест-кейсов завершились успешно, а 15 кейсов отмечены как проваленные из-за несоответствия фактического результата и ожидаемого. Таким образом, доля успешных сценариев составила 75%, доля сценариев с дефектами – 25%, соответственно.

Основные дефекты связаны с нарушением механизмов валидации входных данных, некорректной реализацией бизнес-логики, нарушением принципа единообразия интерфейса и обратной связи пользователю, отсутствием некоторых UI-элементов. Так, допускается создание сущности при отсутствии обязательных полей (локализация боли, дата рождения), отсутствует валидация года рождения, различные форматы сообщений об ошибках в разных формах, отсутствие в некоторых местах подсказок при ошибке и другие. Но самыми критичными являются ошибки бизнес-логики: возможность выполнения действий без выполнения всех обязательных этапов (например, частичное прохождение теста), некорректная логика переходов между этапами регистрации и возможность выполнения операций при некорректных данных.

Фрагмент баг-репорта представлен в таблице 2.

Таблица 2 – Фрагмент баг-репорта

Признак	Описание
Баг – Отсутствие валидации обязательных полей	
ID	BUG-001
Связанный тест-кейс	TC-PAT-001
Severity	Critical
Priority	High
Описание	Система позволяет создать пациента без обязательных полей
Шаги	1 Добавить пациента 2 Не заполнить поля (локализация боли, дата рождения) 3 Сохранить
Ожидаемый результат	Ошибки валидации, создание запрещено
Фактический результат	Пациент создается
Влияние	Некорректные и неполные медицинские данные
Баг – Нарушение бизнес-логики тестирования	
ID	BUG-003
Связанный тест-кейс	TC-VAS-, TC-DN4-, TC-CSI-, TC-HADS-
Severity	Critical
Priority	High
Признак	Описание
Описание	Можно завершить тестирование пациента без прохождения всех его этапов
Шаги	1. Начать тестирование 2. Пропустить/не заполнить любой из тестов (VAS, DN4, CSI, HADS) 3. Завершить тестирование и перейти к результатам
Ожидаемый результат	Все тесты обязательны к прохождению
Фактический результат	Можно завершить без прохождения любого из тестов
Влияние	Некорректная аналитика, неполная информация о состоянии пациента, негативное влияние на постановку диагноза

Заключение. Таким образом, было проведено ручное функциональное тестирование медицинского веб-приложения, предназначенного для автоматизации работы врача с пациентами и проведения медицинских диагностических тестов.

Результаты тестирования показали, что система содержит ряд как существенных, так и не существенных дефектов. К числу наиболее критичных проблем относятся отсутствие строгой валидации входных данных, возможность создания некорректных записей, нарушение последовательности выполнения обязательных сценариев (в частности, прохождения тестов), нехватка функциональности (возможности сбросить фильтры, восстановить пароль), некорректные переходы по гиперссылкам. Кроме этого, были выявлены дефекты, непосредственно связанные с пользовательским интерфейсом: недостаточная информативность системных сообщений, отсутствие единообразного отображения ошибок, некорректная работа отдельных элементов интерфейса.

Проведенное исследование показало, что программное средство требует доработки. В первую очередь, рекомендуется реализовать строгую серверную валидацию данных и доработать механизмы прохождения тестов. Необходимо, чтобы был пройден каждый тест перед расчетом итоговой оценки состояния пациента. Затем необходимо обеспечить корректную обработку ошибок, включая отображение понятных информационных сообщений, унифицировать пользовательский интерфейс, добавить функциональность, которая значительно влияет на удобство использования программного средства (такие функции, как «Восстановление пароля», «Сброс фильтров» и др.), и реализовать корректную навигацию и перенаправления между функциональными модулями приложения.

Проведенная работа подтверждает эффективность использования ручного тестирования на основе тест-кейсов для выявления критических дефектов и оценки качества программных средств на ранних этапах их развития.

Список литературы

1. Ручное тестирование программного обеспечения [Электронный ресурс]. – Режим доступа: <https://apptask.ru/blog/rucnoe-testirovanie-programmnogo-obespecheniia> – Дата доступа: 27.03.2026.

UDC 004.415.53:616.1

**TESTING OF A SOFTWARE SYSTEM FOR THE DIAGNOSIS
AND CORRECTION OF POST-STROKE PAIN SYNDROME**

Shumskaya Y.P., Kushner A.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Kosareva E.M. –master of tech. sciences, assistant of the department of ICSD

Annotation. The functioning of a software tool for the diagnosis and correction of post-stroke pain syndrome has been experimentally investigated. The system is designed to automate patient management processes, perform medical tests (VAS, DN4, CSI, HADS), and generate diagnostic conclusions and treatment recommendations. More than 60 test cases covering key user scenarios were executed, including user registration and authentication, patient management, medical testing, and access control. The percentage of successfully passed test cases was 75%. It was established that the most significant impact on system reliability is caused by defects leading to the creation of incorrect data and violations in the sequence of mandatory testing scenarios. Recommendations for improving the software tool are proposed, including strengthening server-side validation, refining business logic, unifying the user interface, and ensuring correct navigation and redirection between functional components of the system

Keywords: medicine, post-stroke pain syndrome, manual testing, test cases, business logic, REST architecture, medical automation