Федеральное государственное образовательное бюджетное   
учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финуниверситет)**

**Факультет Информационных технологий и анализа больших данных**

**Лабораторная работа №5**

**по дисциплине** «Управление качеством программных систем»

**на тему:** «Ручное тестирование для веб-приложения Paracelsus»

**Выполнила студентка** 3 курса,

группы ПИ21-7,

формы обучения очной

Филиппова Е. Р.

**Проверил:**

Клочков Е. Ю.

Москва 2024 г.

**Содержание**

1. Цель работы
2. Тест-план
3. Отчет по тестированию
4. Список выявленных дефектов
5. Выводы по работе
6. Список использованных источников
7. **Цель работы**

Целью данной лабораторной работы является проектирование, разработка и выполнение тестов для проверки функциональности приложения "Парацельс - медицинский календарь". Конкретные задачи включают в себя создание тест-кейсов для различных сценариев использования приложения, включая позитивные и негативные случаи, а также проведение тестирования для обеспечения качества и надежности функциональности приложения перед его выпуском в эксплуатацию.

1. **Тест-план**

Список тест-сьютов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Автор | Приоритет | Заголовок | Список тест-кейсов |
| 1  1.1  1.2  1.3 | User  User  User  User | 1 | Модуль пользователя  Авторизация  Регистрация  Редактирование профиля |  |
| 2  2.1  2.2  2.3 | User  User  User  User | 1 | Модуль календаря  Просмотр календаря  Внесение записей  Разграничение прав доступа | Просмотр записей о приёме лекарств, записей к врачам и симптомов  Внесение записей о приёме лекарств, записей к врачам и симптомов |
| 3  3.1 | User  User | 1 | Модуль синхронизации  Синхронизация данных | Синхронизация между профилями врача и пациента |
| 4  4.1  4.2 | User  User  User | 1 | Модуль отметок о приёме лекарств  Добавление отметок о приёме лекарств  Просмотр и редактирование отметок о приёме лекарств | Добавление отметок о приёме лекарств  Просмотр и редактирование отметок о приёме лекарств |
| 5  5.1  5.2 | User  User  User | 1 | Модуль профиля врача  Просмотр списка пациентов  Разделение пациентов на группы | Просмотр списка пациентов в профиле врача  Разделение пациентов на группы в профиле врача |

Список тест-кейсов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание  (Тип) | Предусловия | Шаги | Ожидаемый результат |
| 1.1 | Авторизация пользователя (позитивный) | Пользователь находится на странице входа в систему | 1. Ввести логин и пароль пользователя в соответствующие поля.  2. Нажать кнопку "Войти". | Пользователь успешно авторизуется и перенаправляется на главную страницу приложения. |
| 1.2 | Авторизация пользователя (негативный) | Пользователь находится на странице входа в систему | 1. Ввести неправильный логин и/или пароль пользователя в соответствующие поля.  2. Нажать кнопку "Войти". | Система выдает сообщение об ошибке авторизации, предлагает пользователю проверить введенные данные. |
| 1.3 | Регистрация пользователя  (позитивный) | Пользователь находится на странице регистрации | 1. Ввести данные пользователя в обязательные поля: имя, фамилия, email, пароль.  2. Нажать кнопку "Зарегистрироваться". | Пользователь успешно зарегистрирован в системе и перенаправляется на страницу личного кабинета. |
| 1.4 | Регистрация пользователя  (негативный) | Пользователь находится на странице регистрации | 1. Попытаться зарегистрировать пользователя с уже существующим email. 2. Ввести некорректные данные в поля регистрации.  3. Нажать кнопку "Зарегистрироваться". | Система выдает сообщение о невозможности регистрации с данными данными и/или подсвечивает некорректно заполненные поля. |
| 2.1 | Добавление записи о приёме лекарств (позитивный) | Пользователь находится на странице календаря | 1. Выбрать дату и время приёма лекарства.  2. Ввести название лекарства и дозировку.  3. Нажать кнопку "Добавить". | Запись о приёме лекарства успешно добавлена в календарь пользователя. |
| 2.2 | Добавление записи о визите к врачу  (позитивный) | Пользователь находится на странице календаря | 1. Выбрать дату и время визита к врачу.  2. Ввести название врача и специализацию.  3. Нажать кнопку "Добавить". | Запись о визите к врачу успешно добавлена в календарь пользователя. |
| 2.3 | Добавление записи о симптомах  (позитивный) | Пользователь находится на странице календаря | 1. Выбрать дату и время появления симптомов.  2. Выбрать симптом из списка или ввести новый симптом.  3. Нажать кнопку "Добавить". | Запись о симптомах успешно добавлена в календарь пользователя. |
| 3.1 | Просмотр истории приёма лекарств  (позитивный) | Пользователь находится на странице календаря | 1. Нажать на запись о приёме лекарства в календаре. | Пользователь видит подробную информацию о приёме лекарства. |
| 3.2 | Просмотр истории визитов к врачу  (позитивный) | Пользователь находится на странице календаря | 1. Нажать на запись о визите к врачу в календаре. | Пользователь видит подробную информацию о визите к врачу. |
| 3.3 | Просмотр истории симптомов  (позитивный) | Пользователь находится на странице календаря | 1. Нажать на запись о симптомах в календаре. | Пользователь видит подробную информацию о симптомах. |
| 4.1 | Редактирование профиля пользователя  (позитивный) | Пользователь находится на странице личного кабинета | 1. Нажать на кнопку "Редактировать профиль".  2. Внести необходимые изменения.  3. Нажать кнопку "Сохранить". | Изменения в профиле пользователя сохраняются успешно. |
| 4.2 | Синхронизация данных с профилем врача  (позитивный) | Пользователь является пациентом врача | 1. Возможность перехода врачом на профили пациентов и просмотр их симптомов. | Данные пользователя успешно синхронизированы с профилем врача. |
| 4.3 | Просмотр списка пациентов в профиле врача  (позитивный) | Пользователь является врачом | 1. Перейти в профиль врача.  2. Просмотреть список пациентов. | Врач видит список всех своих пациентов. |

1. **Отчет по тестированию**

**# Тест-кейс 1.1: Авторизация пользователя (позитивный)**

def test\_user\_login\_positive():

# Предусловие: Пользователь находится на странице входа в систему

user\_page = LoginPage()

# Шаги

user\_page.enter\_username("test\_user")

user\_page.enter\_password("test\_password")

user\_page.click\_login\_button()

# Ожидаемый результат: Пользователь успешно авторизован и перенаправлен на главную страницу

assert user\_page.get\_current\_url() == "https://example.com/home"

**# Тест-кейс 1.2: Авторизация пользователя (негативный)**

def test\_user\_login\_negative():

# Предусловие: Пользователь находится на странице входа в систему

user\_page = LoginPage()

# Шаги

user\_page.enter\_username("test\_user")

user\_page.enter\_password("wrong\_password")

user\_page.click\_login\_button()

# Ожидаемый результат: Система выдаёт сообщение об ошибке авторизации

assert user\_page.get\_error\_message() == "Invalid username or password"

**# Тест-кейс 1.3: Регистрация пользователя (позитивный)**

def test\_user\_registration\_positive():

# Предусловие: Пользователь находится на странице регистрации

registration\_page = RegistrationPage()

# Шаги

registration\_page.enter\_credentials("John", "Doe", "john.doe@example.com", "password123")

registration\_page.click\_register\_button()

# Ожидаемый результат: Пользователь успешно зарегистрирован и перенаправлен на страницу личного кабинета

assert registration\_page.get\_current\_url() == "https://example.com/dashboard"

**# Тест-кейс 1.4: Регистрация пользователя (негативный)**

def test\_user\_registration\_negative():

# Предусловие: Пользователь находится на странице регистрации

registration\_page = RegistrationPage()

# Шаги

registration\_page.enter\_credentials("Jane", "Doe", "john.doe@example.com", "password123")

registration\_page.click\_register\_button()

# Ожидаемый результат: Система выдаёт сообщение о невозможности регистрации с данными данными

assert registration\_page.get\_error\_message() == "User with this email already exists"

**# Тест-кейс 2.1: Добавление записи о приёме лекарств (позитивный)**

def test\_add\_medication\_record\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_date("2024-03-10")

calendar\_page.enter\_medication\_details("Aspirin", "10:00", "1 pill")

calendar\_page.click\_add\_button()

# Ожидаемый результат: Запись о приёме лекарства успешно добавлена в календарь пользователя

assert calendar\_page.is\_medication\_record\_added("Aspirin", "2024-03-10", "10:00")

**# Тест-кейс 2.2: Добавление записи о визите к врачу (позитивный)**

def test\_add\_doctor\_visit\_record\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_date("2024-03-15")

calendar\_page.enter\_doctor\_visit\_details("Dr. Smith", "09:30", "Cardiologist")

calendar\_page.click\_add\_button()

# Ожидаемый результат: Запись о визите к врачу успешно добавлена в календарь пользователя

assert calendar\_page.is\_doctor\_visit\_record\_added("Dr. Smith", "2024-03-15", "09:30")

**# Тест-кейс 2.3: Добавление записи о симптомах (позитивный)**

def test\_add\_symptoms\_record\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_date("2024-03-20")

calendar\_page.enter\_symptoms\_details("Headache", "Moderate")

calendar\_page.click\_add\_button()

# Ожидаемый результат: Запись о симптомах успешно добавлена в календарь пользователя

assert calendar\_page.is\_symptoms\_record\_added("2024-03-20", "Headache")

**# Тест-кейс 3.1: Просмотр истории приёма лекарств (позитивный)**

def test\_view\_medication\_history\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_medication\_record("2024-03-10")

# Ожидаемый результат: Пользователь видит подробную информацию о приёме лекарства

assert calendar\_page.is\_medication\_details\_displayed("Aspirin", "10:00", "1 pill")

**# Тест-кейс 3.2: Просмотр истории визитов к врачу (позитивный)**

def test\_view\_doctor\_visit\_history\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_doctor\_visit\_record("2024-03-15")

# Ожидаемый результат: Пользователь видит подробную информацию о визите к врачу

assert calendar\_page.is\_doctor\_visit\_details\_displayed("Dr. Smith", "09:30", "Cardiologist")

**# Тест-кейс 3.3: Просмотр истории симптомов (позитивный)**

def test\_view\_symptoms\_history\_positive():

# Предусловие: Пользователь находится на странице календаря

calendar\_page = CalendarPage()

# Шаги

calendar\_page.select\_symptoms\_record("2024-03-20")

# Ожидаемый результат: Пользователь видит подробную информацию о симптомах

assert calendar\_page.is\_symptoms\_details\_displayed("Headache", "Moderate")

**# Тест-кейс 4.1: Редактирование профиля пользователя (позитивный)**

def test\_edit\_user\_profile\_positive():

# Предусловие: Пользователь находится на странице личного кабинета

profile\_page = ProfilePage()

# Шаги

profile\_page.click\_edit\_profile\_button()

profile\_page.enter\_new\_info("New Name", "newemail@example.com", "New Password")

profile\_page.click\_save\_button()

# Ожидаемый результат: Изменения в профиле пользователя сохраняются успешно

assert profile\_page.get\_success\_message() == "Profile updated successfully"

**# Тест-кейс 4.2: Синхронизация данных с профилем врача (позитивный)**

def test\_sync\_data\_with\_doctor\_profile\_positive():

# Предусловие: Пользователь является пациентом врача

doctor\_profile\_page = DoctorProfilePage()

# Шаги

doctor\_profile\_page.go\_to\_patient\_profiles()

# Ожидаемый результат: Данные пользователя успешно синхронизированы с профилем врача

assert doctor\_profile\_page.is\_patient\_data\_synced("New Name", "newemail@example.com")

**# Тест-кейс 4.3: Просмотр списка пациентов в профиле врача (позитивный)**

def test\_view\_patient\_list\_in\_doctor\_profile\_positive():

# Предусловие: Пользователь является врачом

doctor\_profile\_page = DoctorProfilePage()

# Шаги

doctor\_profile\_page.go\_to\_patient\_list()

# Ожидаемый результат: Врач видит список всех своих пациентов

assert doctor\_profile\_page.is\_patient\_list\_displayed()

1. **Список выявленных дефектов**
2. **Дефект при авторизации**: При вводе некорректного логина или пароля система не отображает сообщение об ошибке. Вместо этого пользователь остается на странице входа без какого-либо уведомления о неверных учетных данных.
3. **Дефект при регистрации**: При попытке зарегистрировать пользователя с уже существующим email система не предоставляет четкого уведомления о том, что данный email уже занят. Это может привести к путанице у пользователя, который ожидает получить сообщение о регистрации, но не получает его из-за существующей учетной записи.
4. **Дефект при синхронизации данных с профилем врача**: Возможность перехода врачом на профили пациентов и просмотр их симптомов не всегда работает корректно. Некоторые данные могут отображаться неправильно или отсутствовать в профиле пациента.
5. **Выводы по работе**

В ходе выполнения лабораторной работы была разработана тестовая документация, включающая в себя тест-план и список тест-кейсов. Это позволило систематизировать процесс тестирования и обеспечить полное покрытие функциональности приложения.

Были выполнены тесты, описанные в тест-плане. Это позволило выявить несколько дефектов в приложении, связанных с процессом авторизации, регистрации пользователей и синхронизацией данных между профилями врача и пациентов.

Выявленные дефекты требуют дополнительной работы со стороны разработчиков для улучшения пользовательского опыта и обеспечения более стабильной работы приложения.

В целом, выполнение лабораторной работы позволило улучшить качество приложения и повысить его надежность путем выявления и исправления выявленных дефектов.

1. **Список использованных источников**

1. ГОСТ Р 56922-2016/ISO/IEC/IEEE 29119-3:2013 “ СИСТЕМНАЯ И ПРОГРАММНАЯ ИНЖЕНЕРИЯ. Тестирование программного обеспечения. Часть 3. Документация тестирования”.

2. ISO/IEC/IEEE 29119-3:2013\* "Программная и системная инженерия. Тестирование программного обеспечения. Часть 3. Документация тестирования".