# МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

# УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

# “МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ”

Дисциплина Учебная дисциплина Тестирование и отладка

программного обеспечения

УТВЕРЖДАЮ

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_М.М. Федоськова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЛАБОРАТОРНАЯ РАБОТА № 14**

СОЗДАНИЕ ЧЕК-ЛИСТА ДЛЯ ТЕСТИРОВАНИЯ WEB- ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Разработал преподаватель

Карманов А.В.

2021

Обсуждено и одобрено

на заседании цикловой комиссии спецдисциплин

специальности «Программное обеспечение

информационных технологий»

Протокол № \_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1 Цель работы**

* 1. Сформировать умение создавать чек-лист для тестирования Web- ориентированных приложений

**2 Методическое и материальное обеспечение**

2.1 Методические рекомендации по выполнению лабораторной работы

2.2 Персональный компьютер IBM PC

**3 Последовательность выполнения работы**

3.1 Ознакомиться с теоретическим материалом.

3.2 Выполнить индивидуальное задание.

3.3 Оформить отчет.

1. **Теоретическое обоснование**

Чек-лист тестирования сайта — это список проверок, которые выполняет тестировщик при тестировании веб-сайта.

Во время тестирования веб-приложения нужно обращать внимание на нижеперечисленные пункты. Этот чек-лист применим практически к любому типу веб-приложений в зависимости от бизнес-требований.

Чек-лист для тестирования веб-приложений состоит из:

* тестирования удобства использования.
* функционального тестирования.
* тестирования совместимости.
* тестирования баз данных.
* тестирования безопасности.
* тестирования производительности.

Теперь рассмотрим каждый пункт по отдельности.

**4.1 Тестирование удобства использования**

Это не что иное, как тестирование дружелюбности приложения для пользователя. При тестировании удобства использования проверяется, легко ли новому пользователю разобраться в приложении. В целом при тестировании удобства использования тестируется системная навигация.

Сценарии тестирования удобства использования:

* содержание веб-страницы верное, без грамматических и орфографических ошибок.
* все шрифты соответствуют требованиям.
* все тексты правильно выровнены.
* все сообщения об ошибках верные, без орфографических и грамматических ошибок, и соответствуют заголовку окна.
* подсказки существуют для всех полей.
* все поля правильно выровнены.
* между полями, колонками, рядами и сообщениями об ошибках оставлено достаточно свободного места.
* все кнопки должны иметь стандартный формат и размер.
* ссылка на домашнюю страницу должна быть на каждой странице сайта.
* неактивные поля должны быть серыми.
* проверьте, что на сайте нет битых ссылок и изображений.
* подтверждающие сообщения должны отображаться для всех операций обновления и удаления.
* проверьте сайт при разных разрешениях экрана ((640 x 480, 600x800 и т. д.)
* проверьте, что пользователь может пользоваться системой без раздражения.
* проверьте, что tab правильно работает.
* панель скролла должна появляться только тогда, когда она требуется.
* если при отправке формы есть сообщения об ошибке, в нем должна содержаться информация, переданная пользователем.
* заголовок должен отображаться на каждой странице.
* все поля (текстовые, выпадающие меню, радио-кнопки и т. д.) и кнопки должны быть доступны с клавиатуры, и пользователь должен быть в состоянии пользоваться сайтом, используя только клавиатуру.
* убедитесь, что данные в выпадающих списках не обрезаются из-за размеров поля, и проверьте, зашиты ли данные в код или управляются администратором.

**4.2 Функциональное тестирование**

Тестирование функциональностей и операционного поведения продукта с целью убедиться, что они соответствуют спецификациям. Тестирование, игнорирующее внутренние механизмы системы или компонента. Оно концентрируется исключительно на выходных данных, полученных в ответ на пользовательский ввод и условия исполнения сценариев.

Цель функционального тестирования – убедиться, что ваш продукт соответствует нужной функциональной спецификации, упомянутой в вашей документации по разработке.

Сценарии функционального тестирования:

* протестируйте валидацию всех обязательных полей
* убедитесь, что знак звездочки отображается у всех обязательных полей
* убедитесь, что система не отображает окно ошибки при незаполненных необязательных полях.
* убедитесь, что високосные коды корректно валидируются и не вызывают ошибок в расчетах.
* протестируйте числовые поля: они не должны принимать буквы, в этом случае должно отображаться соответствующее сообщение об ошибке.
* протестируйте отрицательные значения в числовых полях, если они разрешены.
* протестируйте, что деление на ноль верно обсчитывается.
* протестируйте максимальную длину каждого поля, чтобы убедиться, что данные не обрезаются.
* протестируйте всплывающее сообщение ("это поле ограничено 500 знаками"), которое должно отобразиться, если введенные данные превышают разрешенный размер поля.
* убедитесь, что подтверждающее сообщение отображается для операций обновления и удаления.
* убедитесь, что значения стоимости отображаются в нужной валюте.
* протестируйте все поля ввода на спецсимволы.
* протестируйте функциональность тайм-аута.
* протестируйте функциональность сортировки.
* протестируйте функциональность доступных кнопок.
* протестируйте условия использования и часто задаваемые вопросы: они должны быть внятными и доступными пользователю.
* протестируйте, что при отказе функциональности пользователь перенаправляется на специальную страницу ошибки.
* протестируйте, что все загруженные документы правильно открываются.
* протестируйте, что пользователь может скачать загруженные файлы.
* протестируйте почтовую функциональность системы.
* протестируйте, что java script верно работает в разных браузерах (ie, firefox, chrome, safari, opera).
* посмотрите, что будет, если пользователь удалит куки, находясь на сайте.
* посмотрите, что будет, если пользователь удалит куки после посещения сайта.
* протестируйте все данные в выпадающих списках: они должны быть расположены в хронологическом порядке.

**4.3 Тестирование совместимости**

Тестирование совместимости используется, чтобы убедиться, что ваше приложение совместимо с другими элементами системы, в которой оно работает – например, браузерами, операционными системами или железом.

Цель тестирования совместимости – оценка того, насколько хорошо ПО работает в определенном браузере, под определенной ОС, с другим ПО или железом.

Сценарии тестирования совместимости:

* протестируйте сайт в различных браузерах (IE, Firefox, Chrome, Safari, Opera) и убедитесь, что сайт правильно отображается.
* убедитесь, что используемая версия HTML совместима с соответствующими версиями браузеров.
* убедитесь, что картинки корректно отображаются в разных браузерах.
* убедитесь, что шрифты верно отображаются в разных браузерах.
* убедитесь, что Java Script код работает в разных браузерах.
* проверьте анимированные GIF в разных браузерах.

**4.4 Тестирование баз данных**

При тестировании баз данных проверяются бэкэнд-записи, введенные через веб или десктоп-приложение. Данные, которые отображаются в приложении, должны совпадать с данными, хранящимися в базе данных.

Чтобы тестировать базы данных, тестировщик должен знать следующее:

* тестировщик должен понимать функциональные требования, бизнес-логику, основной сценарий приложения и дизайн базы данных.
* тестировщик должен разбираться в таблицах, триггерах, процедурах хранения, способах отображения и указателях, используемых для приложения.
* тестировщик должен понимать логику триггеров, процедур хранения, способов отображения и указателей.
* тестировщик должен понимать, какие таблицы затрагиваются, когда операции вставки, обновления и удаления выполняются в приложении.

Сценарии тестирования баз данных:

* проверьте название базы данных: оно должно совпадать со спецификацией.
* проверьте таблицы, колонки, типы колонок и значения по умолчанию: все это должно совпадать со спецификацией.
* проверьте, позволяет ли колонка значение null.
* проверьте первичный и внешний ключ каждой таблицы.
* проверьте процедуры хранения.
* протестируйте, установлена ли процедура хранения.
* проверьте название процедуры хранения.
* проверьте названия параметров, их типы и количество.
* проверьте, обязательны параметры или нет.
* проверьте процедуру хранения, удалив некоторые параметры.
* проверьте базу данных, если на выходе ноль – записи с нулем должны быть задействованы.
* проверьте процедуру хранения, задав простые sql-запросы.
* убедитесь, что процедура возвращает значения.
* проверьте процедуру вводом тестовых данных.
* проверьте поведение каждого флага в таблице.
* убедитесь, что данные правильно сохраняются в базе данных после каждого ввода.
* проверьте данные при каждой операции обновления, удаления и вставки.
* проверьте длину каждого поля. длина на бэкэнде и фронтэнде должны совпадать.
* проверьте названия баз данных qa, uat и прода. имена должны быть уникальными.
* проверьте зашифрованные данные в базе.
* проверьте размер базы и время отклика на каждый запрос.
* проверьте данные, отображающиеся на фронтэнде, и убедитесь, что они совпадают с бэкэндом.
* проверьте целостность данных, вводя невалидные значения в базу.
* проверьте триггеры.

**4.4 Тестирование безопасности.** Тестирование безопасности нацелено на поиск недостатков и пробелов с точки зрения безопасности приложения.

Сценарии тестирования безопасности:

* убедитесь, что страницы, содержащие важные данные (пароль, номер кредитной карты, ответы на секретные вопросы и т. п.) открываются через https (ssl).
* убедитесь, что важная информация (пароль, номер кредитной карты) отображается в зашифрованном виде.
* убедитесь, что правила создания паролей внедрены на всех страницах авторизации (регистрация, страница "забыли пароль", смена пароля).
* убедитесь, что если пароль изменен, пользователь не может зайти под старым.
* убедитесь, что сообщения об ошибках не содержат никакой секретной информации.
* убедитесь, что если пользователь вышел из системы или сессия завершена, он не может пользоваться сайтом.
* проверьте доступ к закрытым и открытым страницам сайта напрямую без авторизации.
* убедитесь, что опция "просмотр исходного кода" отключена и не видна пользователю.
* убедитесь, что учетная запись пользователя блокируется, если он несколько раз ввел пароль неверно.
* убедитесь, что пароль не хранится в куки.
* убедитесь, что если какая-либо функциональность не работает, система не отображает информацию о приложении, сервере или базе данных. вместо этого отображается соответствующее сообщение об ошибке.
* проверьте сайт на sql-инъекции.
* проверьте права пользователей и их роли. к примеру, кандидат не должен быть способен получить доступ к странице администратора.
* убедитесь, что важные операции пишутся в логи, и информацию можно отследить.
* убедитесь, что значения сессий отображаются в адресной строке в зашифрованном виде.
* убедитесь, что куки хранятся в зашифрованном виде.
* проверьте приложение на устойчивость к брутфорс-атакам.

**4.5 Тестирование производительности**

Тестирование производительности проводится для оценки соответствия системы или компонента специфичным требованиям к производительности.

Общие тестовые сценарии:

* определение производительности, стабильности и масштабируемости приложения под разной нагрузкой.
* определение, может ли актуальная архитектура поддерживать приложение при пиковых нагрузках.
* определение, какая конфигурация приводит к наилучшим показателям производительности.
* определение бутылочного горла приложения и инфраструктуры.
* определение, не изменилось ли время отклика у новой версии приложения.
* оценка продукта и/или железа с целью удостовериться, что они выдержат прогнозируемые объемы нагрузки.

**5 Индивидуальное задание**

Составить чек-лист для тестирования выбранного web-приложения

**6 Содержание отчета** (в электронном виде, в рабочей папке учащегося)

6.1 Тема работы.

6.2 Цель работы.

6.3 Разработанные тесты

**Список используемых источников**

[2], c. 186-193