Тестирование web-приложения — frontend АС Verita (СУМД)

# тестовые случаи

1. запуск приложения
2. анонимный пользователь создает новое пространство имен
3. просмотр существующих пространств имен
   1. позитивный
      1. анонимный пользователь обращается к открытому пространству имен
   2. негативный
      1. анонимный пользователь обращается к закрытому пространству имен
4. вход в систему
   1. позитивный
      1. правильное имя,пароль
   2. негативный
      1. неправильное имя,пароль
         1. сообщение
         2. исключение
5. просмотр существующих пространств имен
   1. позитивный
      1. анонимный пользователь обращается к открытому пространству имен
      2. авторизованный пользователь с правами на пространство имен
   2. негативный
      1. анонимный пользователь обращается к закрытому пространству имен
      2. авторизованный пользователь без прав на пространство имен
6. поиск пространства имен, конкретного объекта
   1. позитивный
      1. правильное наименование
   2. негативный
      1. неправильное наименование
7. создание нового пространства имен
   1. позитивный
      1. все поля заполнены
      2. заполнены только обязательные поля
   2. негативный
      1. все поля не заполнены
      2. не заполнены обязательные поля

**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.

**Сценарии тестирования безопасности:**

1. Убедитесь, что страницы, содержащие важные данные (пароль, номер кредитной карты, ответы на секретные вопросы и т. п.) открываются через HTTPS (SSL).
2. Убедитесь, что важная информация (пароль, номер кредитной карты) отображается в зашифрованном виде.
3. Убедитесь, что правила создания паролей внедрены на всех страницах авторизации (регистрация, страница "забыли пароль", смена пароля).
4. Убедитесь, что если пароль изменен, пользователь не может зайти под старым.
5. Убедитесь, что сообщения об ошибках не содержат никакой секретной информации.
6. Убедитесь, что если пользователь вышел из системы или сессия завершена, он не может пользоваться сайтом.
7. Проверьте доступ к закрытым и открытым страницам сайта напрямую без авторизации.
8. Убедитесь, что опция "Просмотр исходного кода" отключена и не видна пользователю.
9. Убедитесь, что учетная запись пользователя блокируется, если он несколько раз ввел пароль неверно.
10. Убедитесь, что пароль не хранится в куки.
11. 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.
12. Проверьте сайт на SQL-инъекции.
13. Проверьте права пользователей и их роли. К примеру, кандидат не должен быть способен получить доступ к странице администратора.
14. Убедитесь, что важные операции пишутся в логи, и информацию можно отследить.
15. Убедитесь, что значения сессий отображаются в адресной строке в зашифрованном виде.
16. Убедитесь, что куки хранятся в зашифрованном виде.
17. Проверьте приложение на устойчивость к брутфорс-атакам.

Функциональное тестирование веб-приложений

* Проверка внешних и внутренних ссылок
* Тестирование интерфейса (UI) – вёрстка и поведение
* Тестирование бизнес-логики (это, собственно, и есть "настоящее" тестирование :))
* Тестирование навигации – кнопки "вперёд/назад" и "перезагрузить страницу"

# тестовые наборы

# вид и объем тестирования

1. Функциональное тестирование. Это тестирование используется для проверки всех ссылок веб-страниц; Тестирования форм, файлов cookie и подключение к базе данных.

2. Тестирование удобства использования: это тестирование проверяет навигацию и удобство пользования веб-страницами. Благодаря этому тестированию гарантируется, правильность восприятия содержимого конечным пользователем. Оно также проверяет, правильно ли работают текстовые ссылки привязки, имеют ли файлы файлы справки необходимую информацию и все ли ссылки работают.

3. Тестирование интерфейса. Проверяет, правильно ли взаимодействуют интерфейс веб-сервера и сервера приложений, сервер приложений и сервер баз данных. Этот тест гарантирует, что пользователи не будут видеть никаких сообщений об ошибках.

4. Тестирование совместимостb. Тестирование на совместимость очень важно, так как оно проверяет совместимость браузера, совместимость с операционной системой, параметры браузера и печати.

5. Тестирование производительности. Тестирование производительности включает в себя тестирование веб-нагрузки и стресс-тестирование сети. Тестирование веб-загрузки проверяет, могут ли многие пользователи одновременно получить доступ к одной и той же странице, и может ли веб-страница обрабатывать большую нагрузку на какой-либо конкретной странице. Веб-стресс-тестирование проводится на сайте, чтобы увидеть, как будет реагировать сайт и восстанавливаться во время стресса.

6. Тестирование безопасности. Проверяет безопасность веб-приложений. В целях безопасности внутренние страницы не должны открываться, если вы не вошли на веб-сайт. Другая статистика не должна отображаться, даже если пользователь вошел в систему. Файлы должны быть доступны только при входе в аккаунт, и к ним нельзя получить доступ без этого. CAPTCHA для автоматизации входа должна быть также протестирована. SSL должен быть протестирован на меры безопасности.

## сокращенный тестовый набор — dry run, smoke test

1. Объем
   1. запуск приложения
   2. анонимный доступ к существующему пространству имен (позитивный тест)
   3. авторизация в системе (позитивный тест)
   4. создание нового пространства имен
   5. удаление пространства имен
   6. выход из системы

## функциональное тестирование — для конкретной функции или их набора

## полное тестирование

## регрессионное тестирование