GeekBrains

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА по курсу «Ручное тестирование»

ТЕСТИРОВАНИЕ WEB-ПРИЛОЖЕНИЯ

Выполнил:

студентка курса «ручное тестирование»

Кудряшова Алена Николаевна

2022

# Содержание

Введение в тестирование 1

Чек-лист 2

Виды чек-листов

Преимущества и недостатки чек-листов

Создание чек-листа по ТЗ 3

Тестирование по чек-листу 4

Тест-кейс 5

Преимущества и недостатки тест-кейсов

Создание тест-кейса по ТЗ 6

Тестирование по тест-кейсу 7

Техники тест дизайна ..................................8

Виды тестирования 9

Создание отчета о тестировании 10

Заключение 11

Список иллюстраций ii

Словарь терминов 12

Список использованных источников 13

### Введение в тестирование

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом (ISO/IEC TR 19759:2005).

В разное время и в различных источниках тестированию давались различные определения, в том числе:

* процесс выполнения программы с целью нахождения ошибок;
* интеллектуальная дисциплина, имеющая целью получение надежного программного обеспечения без излишних усилий на его проверку;
* техническое исследование программы для получения информации о её качестве с точки зрения определённого круга заинтересованных;
* проверка соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выполненных определённым образом;
* процесс наблюдения за выполнением программы в специальных условиях и вынесения на этой основе оценки каких-либо аспектов её работы;
* процесс, имеющий целью выявление ситуаций, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации;
* процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.

ЧЕК-ЛИСТ

Чек-лист - список, содержащий ряд необходимых проверок для какой-либо работы.

Важность чек листов трудно переоценить. Каким бы опытным ни был сотрудник, в спешке он может легко забыть важную деталь.

В тестировании чек-лист — это список проверок для тестирования продукта. Чек-листы устроены предельно просто. Любой из них содержит перечень блоков, секций, страниц, других элементов, которые следует протестировать, например:

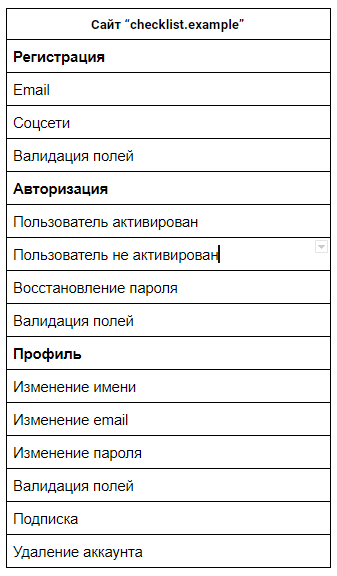


Рисунок 1 – Пример чек листа

Выполненные пункты отмечаются статусами, например: “Passed”, “Failed”, “Blocked”, “Skipped”, “Not run”. Эти статусы также могут иметь свой цвет:

C:\Users\user\Documents\GraduateWorkKudryashova\2.png

Рисунок 2 - Статусы

Преимущества использования чек-листов:

* улучшить представление о системе в целом, видеть статус ее готовности;
* понимать объем проделанной и предстоящей работы по тестированию;
* не повторяться в проверках и не упустить ничего важного в процессе тестирования.

ВИДЫ ЧЕК-ЛИСТОВ

Можно выделить два вида чек-листов: специальные и универсальные.

Специальные чек-листы создаются и используются для конкретных проектов, поэтому пункты такого чек-листа соответствуют специфики проекта. Тестировщик по специальному чек-листу проверяет возможность выполнить уникальное действие, предусмотренное требованиями. Вот примеры пунктов специального чек-листа:

* при наведении курсора на пункт меню “Товары”, должен меняться цвет на синий. Указатель должен менять форму на pointer;
* если пользователь открыл страницу “Ваша корзина” и в корзине присутствует хотя бы один товар, то должно показываться уведомление.

Такие чек-листы не подходят к использованию на других проектах.

Универсальные чек-листыподходят для тестирования проектов одного типа. Проверка по универсальному чек-листу не привязывается к графическим элементам или конкретной реализации, а проверяется сама возможность пользователя выполнить действие. Для универсального чек-листа составляется абстрактный список проверок. Пункты универсального чек-листа могут быть такими:

* пользователь может перейти в раздел “Товары”;
* оплата должна совершаться;
* товар должен добавляться в корзину;
* ссылки при наведении подчеркиваются;
* валидатор верстки показывает отсутствие ошибок и т.п.

Универсальные чек-листы можно использовать повторно на проектах одного типа. У многих агентств есть такие универсальные чек-листы, по ним определяется общий уровень качества продукта.

Чтобы составить работающий чек-лист, обратите внимание на эти рекомендации:

1. Один пункт = одна проверка. Минимальная полная операция проводимая тестировщиком при проверке — это один пункт чек-листа:
2. При составлении чек-листа нужно опираться на требования, чтобы не тестировать то, что не существенно.
3. Давайте пунктам чек-листа названия по форме, общей для всех членов команды, чтобы работа с чек-листом не вызывала неоднозначных толкований. Можно договориться использовать во всех пунктах только глаголы в инфинитиве или существительные: «проверить»/ «добавить»/ «отправить» либо «проверка»/«отправка»/«добавление».
4. Детализируйте чек-лист в зависимости от задачи.
5. Объединяйте чек-листы в матрицы, где можно отразить не только сами проверки, но и условия проверки (платформа, версия продукта, сотрудник и т.п.) и статус проверки. Матрицы — это компромисс между чек-листами и тест-кейсами. Их легче поддерживать, чем тест-кейсы, так как в такой таблице отсутствуют шаги (steps). В них одна строка = одна проверка:

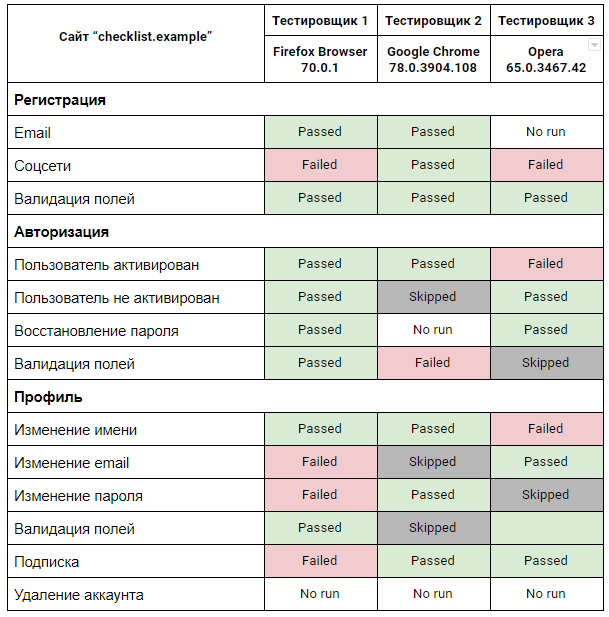


Рисунок 3 - Матрица

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ЧЕК-ЛИСТОВ

Преимущества:

* чек-лист легко читается;
* по чек-листу быстро тестировать: в тест-кейсе нужно отмечать статус каждого шага, в то время как в чек-листе достаточно одной строчки;
* чек-лист — источник результатов для отчёта: можно быстро посчитать сколько проверок выполнено, и в каком они статусе, узнать количество открытых репортов;
* в любой момент можно узнать статус — всегда есть то, что нужно проверить в первую очередь, можно упорядочить пункты чек-листа или изменить порядок, когда это требуется.

Недостатки:

* неопределенность тестового набора: каждый тестировщик выполняет пункт чек-листа по-своему;
* неопределенность тестовых данных;
* недостаточность детализации;
* сложнее обучить начинающих сотрудников: пункты чек-листа чаще абстрагируются от конкретных элементов интерфейса и описывают то, что нужно сделать;
* чек-лист менее эффективен для начинающих тестировщиков, лучше использовать тест-кейсы.

Чек-листы лучше применять на ранних этапах, когда софт быстро меняется, потому что тест-кейсы дорого поддерживать.

СОЗДАНИЕ ЧЕК-ЛИСТА ПО ТЗ

Создадим чек-лист на примере сайта <https://limelab.tech/foodbuzz>.

Специальные чек-листы создаются и используются для конкретных проектов, поэтому пункты такого чек-листа должны соответствовать ТЗ и независимо от реализации. Для примера берем раздел сайта «Футер» и читаем его описание в документации к сайту.

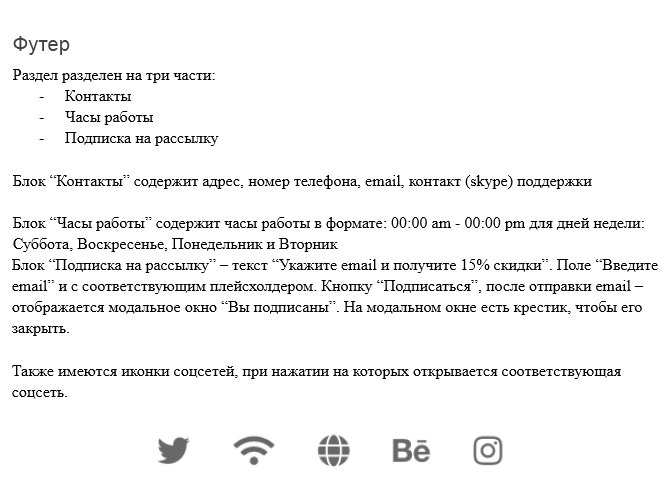


Рисунок 4 - Описание из ТЗ

Составляем чек-лист по принципу «Один пункт = одна проверка».

Условно разделим проверку на 4 части: контакты, часы работы, подписка на рассылку и иконки соцсетей. Каждую часть разделим на проверки.

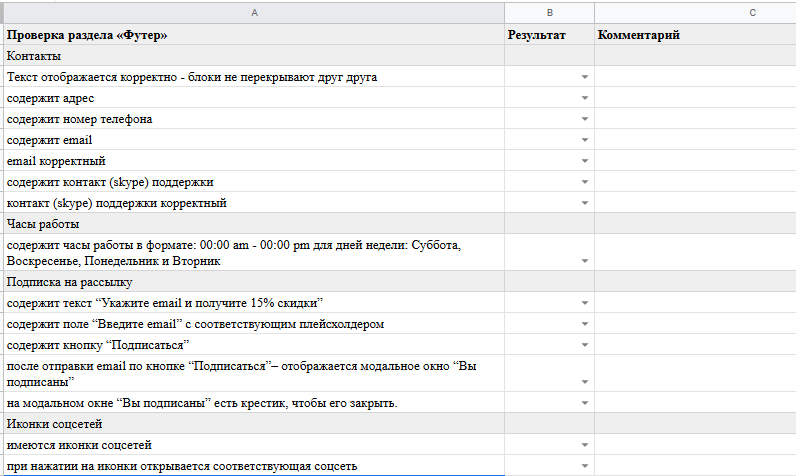


Рисунок 5 - Чек-лист раздела "Футер"

ТЕСТИРОВАНИЕ ПО ЧЕК-ЛИСТУ

Устанавливаем результат проверки для каждого пункта и пишем комментарии для статусов “Failed”, “Blocked”, “Skipped”.

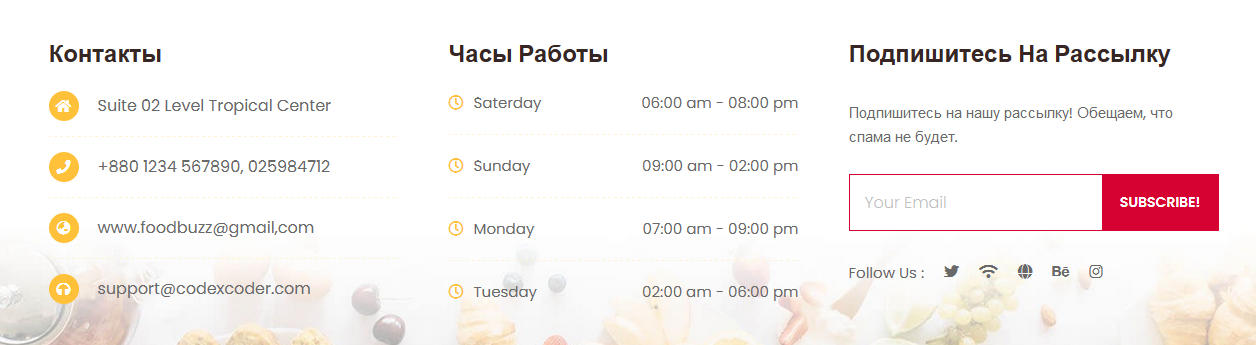


Рисунок 6 - Раздел "Футер"

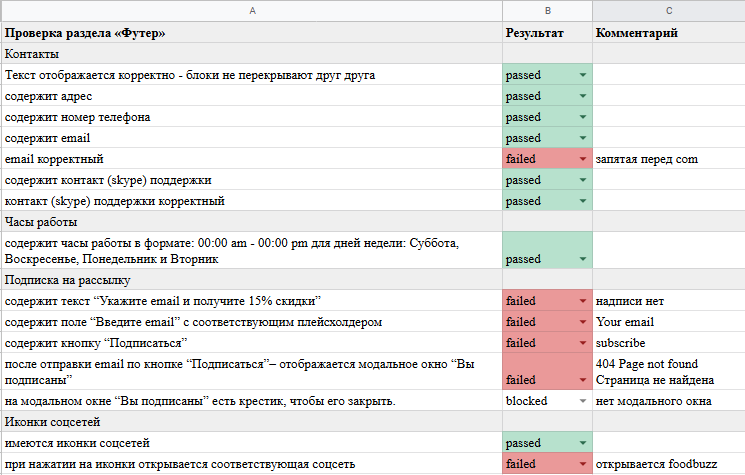


Рисунок 7 - Чек-лист для раздела "Футер"

ТЕСТ-КЕЙС

Тест-кейс — это проверка. "Выполни тест-кейс по вводу отрицательных значений" = проведи проверку такую-то и проверь, что результат будет такой-то.

Тест-кейс — это описание проверки работы системы, которое может выполнить любой человек из команды, будь то тестировщик, разработчик, аналитик или даже бизнес-заказчик.

Набор тест-кейсов называется тестовым набором (test suite).

Иногда этот набор некорректно называют тест-планом. Тест-план — это именно план: когда, что, зачем, какими ресурсами.

Стандартные атрибуты тест-кейса:

* Номер — уникальный идентификатор тест-кейса. Его удобно использовать для одинакового понимания, о какой проверке идет речь (например, дать ссылку в баге).
* Название — краткое описание сути проверки. Должно помещаться в твиттер и быть понятным! Кратко, но емко.
* Предварительные шаги — описание действий, которые необходимо выполнить, но прямого отношения к проверке они не имеют (например, зарегистрироваться в системе для проверки создания элемента). Если предварительных шагов нет, то секция не заполняется.
* Шаги — описание действий, необходимых для проверки (например, создание элемента).
* Ожидаемый результат (ОР) — сама проверка: что мы ожидаем получить после выполнения шагов ("Элемент создан").

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ТЕСТ-КЕЙСОВ

Преимущества: тест-кейсы можно доверить выполнять новичку или призванному на помощь коллеге из другого отдела, который ничего о проекте не знает. Дополнительных вопросов с его стороны будет по минимуму — все должно быть понятно.

Недостатки:

1. Очень много копипасты. Например при заполнении поля "ФИО". Тест-кейсы «ввести в поле только символы, только числа, строку нулевой длины и т. д.» будут очень похожи друг на друга, первые шаги одинаковые и будут копипаститься.
2. Сложно поддерживать. Представьте, что вкладку «Жильцы» переименовали в «Заказчики». Чтобы актуализировать тест-кейсы, надо внести изменения в сотни сценариев, что утомительно даже в режиме "Ctrl + C, Ctrl + V".
3. Неактуальное состояние. Тест-кейсы копипастятся друг от друга, и часто в них остаются неактуальные части из исходного кейса, которые забыли изменить.

СОЗДАНИЕ ТЕСТ-КЕЙСА ПО ТЗ

Возьмем сайт и раздел как для чек-листа.

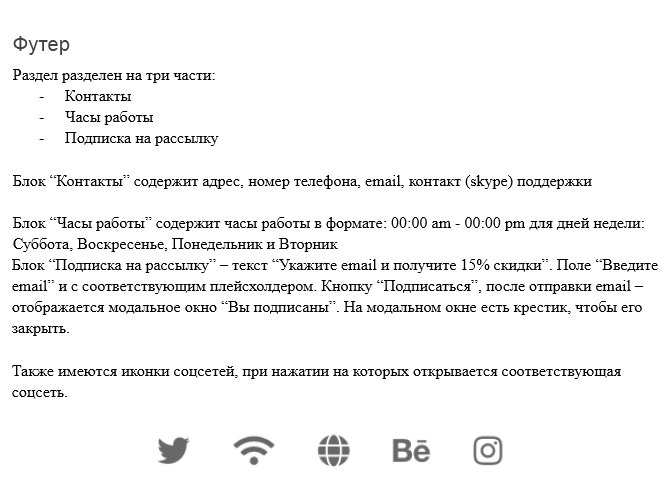


Рисунок 8 - Описание из ТЗ

Создадим тест-кейс используя стандартные атрибуты.

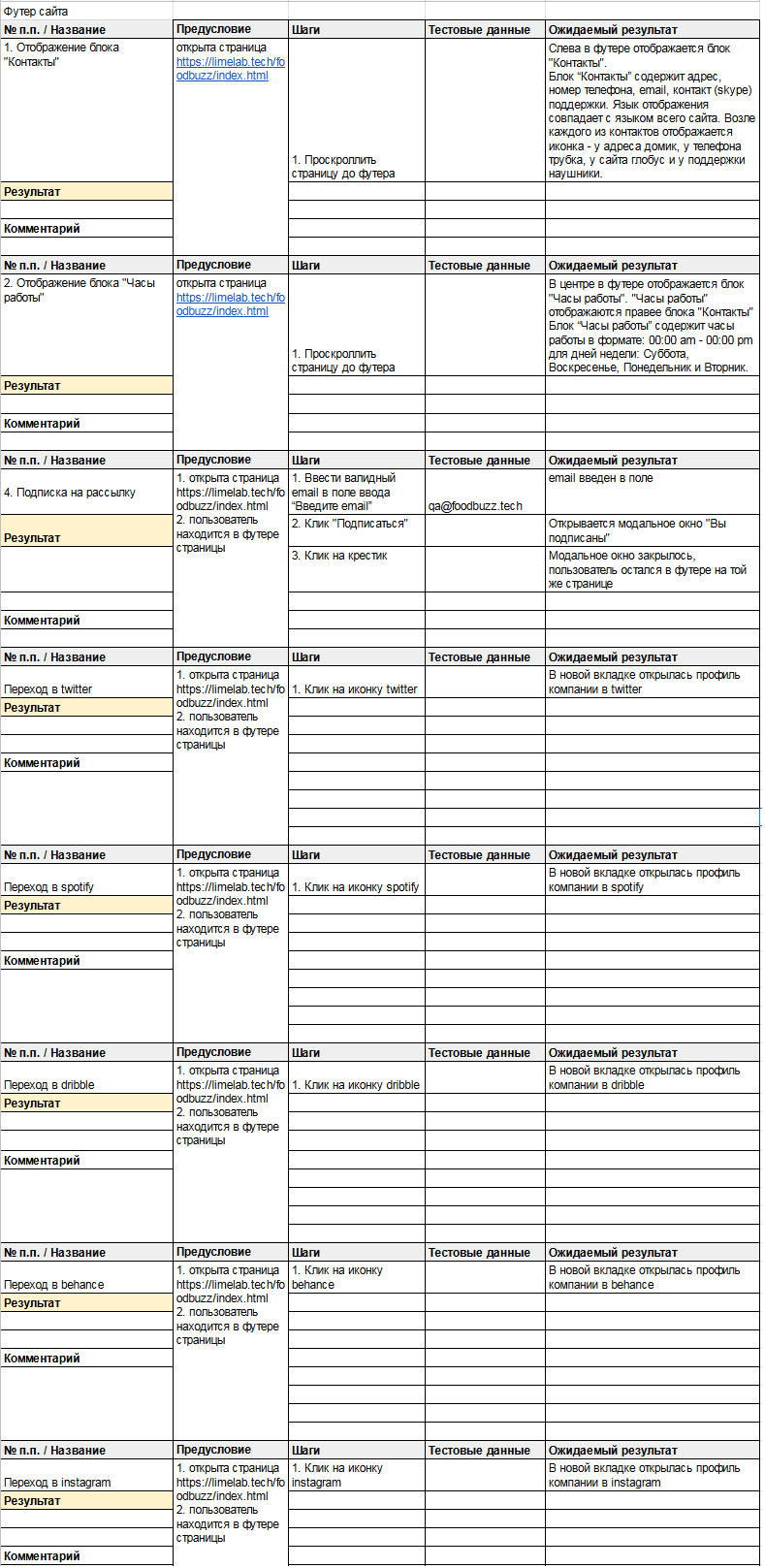


Рисунок 9 - Тест-кейс для раздела "Футер"

ТЕСТИРОВАНИЕ ПО ТЕСТ-КЕЙСУ

Выполняем проверку, используя предусловие и тестовые данные, выполняя действия из раздела «Шаги». Устанавливаем результат проверки для каждого пункта тест-кейса и пишем комментарии для статусов “Failed”, “Blocked”, “Skipped”.

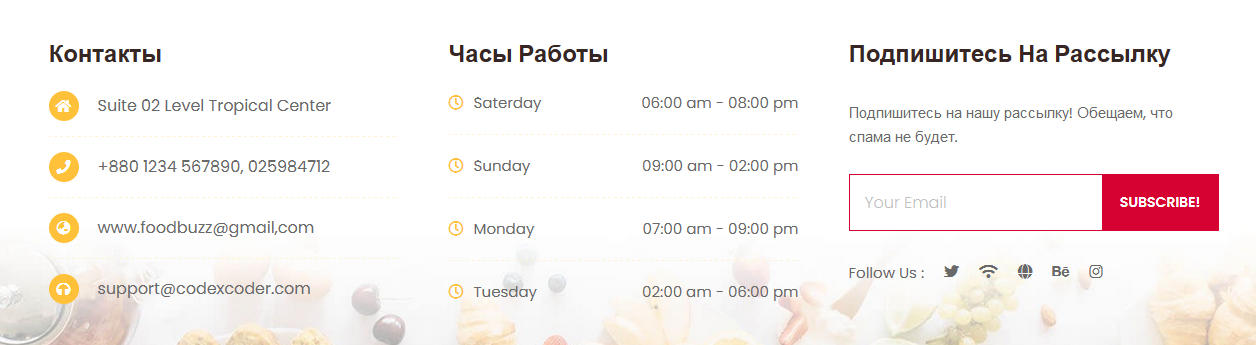


Рисунок 10 - раздел "Футер"

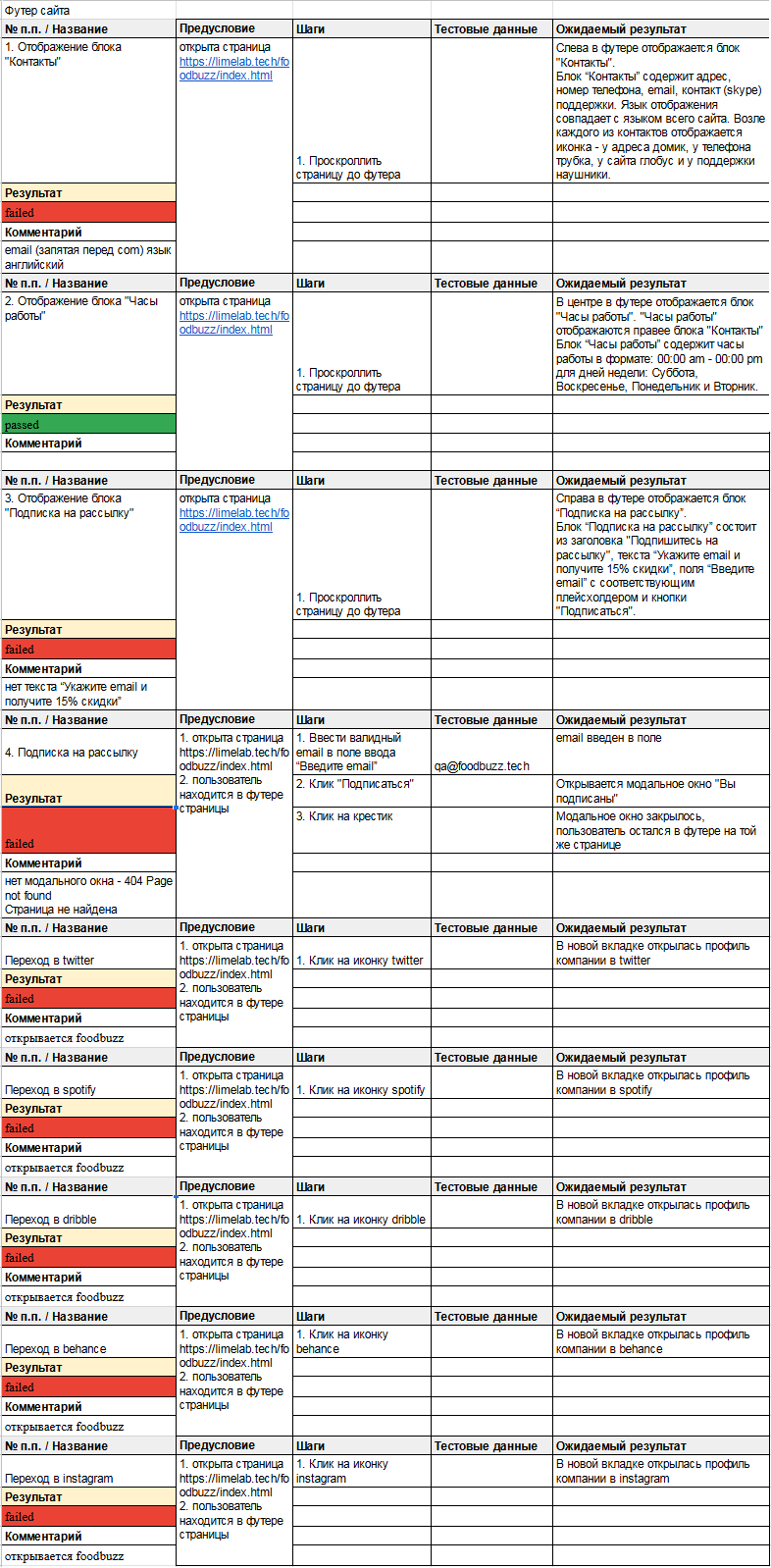


Рисунок 11 - Тест-кейс для раздела "Футер"

# Список иллюстраций

Номер п/п Страница

1. Рисунок 12 – Пример чек листа 1
2. Рисунок 13 – Статусы 2
3. Рисунок 14 – Матрица 3

# Словарь терминов

Тести́рование програ́ммного обеспе́че́ния. процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* 1. <https://gb.ru/>
  2. <https://qualitica.ru/blog/chek-list/>
  3. <https://software-testing.ru/library/testing/testing-for-beginners/1991-test-case-writing>