# TODO-list

## Введение

Данный тест-план описывает процесс тестирования приложения Todo-list.

Todo-list - проект со списками ведения дел по принципу "todo". Позволяет добавлять, изменять, удалять задачи и выводить их списком с возможностью сортировки.

Целью тестирования приложения Todo-list является проверка корректности работы его функционала и заявленных требований.

Итогом процесса тестирования должны стать написанные автотесты и отчет, дающий представление о текущем качестве продукта.

Тестирование будет производиться с помощью фреймворка Codeception, тесты будут написаны на языке PHP.

## Нужно проверить

* ручная проверка, что приложение хранит данные локально - для каждого браузера свои
* приемочное автотестирование - проверяем работу основного функционала
* только позитивные сценарии

## Не проверяется

* визуальная регрессия - тестировать скриншотами не будем, хватит предварительного ручного тестирования
* нагрузочное - проверим только на примере одного пользователя
* негативные сценарии и требования с низким приоритетом - под них не заложено время

## Требования

Компоненты в PC-версии:

1. Main page
2. Todo-card
3. Table header
4. Table row
5. Deleting window

Компоненты в mobile-версии (отличающиеся):

1. Table header
2. Sorting button

Требования по функционалу:

1. приложение работает с локальными данными
2. существующие todo отображаются в таблице
3. добавление нового todo
4. изменение существующего todo
5. удаление существующего todo
6. удаление всех todos разом
7. окно изменения карточки todo можно закрыть без изменений
8. окно добавления карточки todo можно закрыть без изменений
9. окно удаления todo можно закрыть без изменений
10. окно удаления всех todo можно закрыть без изменений
11. при нажатии вне окна (modal-background) - окно закрывается без изменений
12. при создании карточки поля должны быть пустыми
13. todo обязано иметь имя
14. имя todo не обязано быть уникальным
15. todo не обязано иметь приоритет
16. todo может иметь приоритет: important - высший, secondary - средний, meh - низкий
17. сортировка todo сохраняется после обновления страницы
18. сортировка по названию
19. сортировка по приоритету

## Стратегия тестирования

1. Написать автотесты, проверяющие функционал приложения.
2. Убедиться, что в каждом браузере хранятся свои данные веб-приложения.

## Условия начала

* проведено первичное ручное тестирование
* баги и замечания зафиксированы в отчете

## Условия завершения

* все пункты из блока "Нужно проверить" выполнены
* баги и замечания зафиксированы в отчете

## Сроки

* планирование тестирования - 3ч
* изучение продукта - 0,5ч
* анализ требований - 1,5ч
* написание тест-кейсов - 3ч
* написание и настройка тестового фреймворка - 5ч
* написание автотестов - 8ч
* проверка соответствия полученных автотестов требованиям - 1ч
* отчет о тестировании - 3ч

Итого: 25 часов

## Тестовая среда

* браузеры chrome 79.0.3945.130 64-bit и firefox 72.0.2 64-bit
* драйверы chromedriver 79.0.3945.36 и geckodriver v0.26.0
* разрешения экрана 1366x768, 360x640

## Риски

* тестовый сайт не будет доступен
* написание автотестов займет больше ожидаемого времени
* критические баги помешают проверке функционала