



ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ.
Государственное образовательное учреждение высшего профессионального
образования.
**«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» имени В.И. Ульянова (Ленина)»
(СПБГЭТУ)**

УТВЕРЖДЁН

обозначение листа утверждения

Recommendation Applicant Service

Название проекта

TEST PLAN / ПЛАН ПО ТЕСТИРОВАНИЮ

наименование документа

C09_RAS_STP.doc

Электронный / бумажный

вид носителя данных

6 страниц

объём документа

Санкт-Петербург
2021

ИСТОРИЯ ВЕРСИЙ

Версия #	Кто разработал	Дата	Причина
00.10	Фомичев Д.А. Паушев Коробейников Р.А. Вдовиченко	25.02	Первый вариант описания содержания проекта

Список исполнителей

№	ФИО	Группа	Обязанности	Контактные данные (email/телефон)
1	Фомичев Дмитрий Алексеевич	6308	ML, data analyst	savior.7@yandex.ru
2	Коробейников Роман Андреевич	6308	ML, tester	romka.kra@mail.ru
3	Паушев Дмитрий Александрович	6308	Frontend	mpaushev2@mail.ru
4	Вдовиченко Валерия Владимировна	6308	ML	lervd@mail.ru

Преподаватель

Филиппов Евгений Васильевич e.philippov.leti@mail.ru

Бурукин Станислав SBurukin@luxoft.com

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ	4
1.1 НАЗНАЧЕНИЕ ДОКУМЕНТА	4
1.2 ЦЕЛЬ ТЕСТИРОВАНИЯ.....	4
2.1 ВИДЫ ТЕСТИРОВАНИЯ	4
2.1.1 Тестирование кроссплатформенности	4
2.1.2 Тестирование моделей машинного обучения	5
2.1.3 Тестирование пользовательского интерфейса	5
2.1.4 Тестирование производительности	5
Тестирование кроссплатформенности	6
Тестирование моделей машинного обучения	6
Тестирование пользовательского интерфейса	6
Тестирование производительности	6

1 ВВЕДЕНИЕ

1.1 НАЗНАЧЕНИЕ ДОКУМЕНТА

Целью настоящего тест плана является описание планирования процесса тестирования.

1.2 ЦЕЛЬ ТЕСТИРОВАНИЯ

Цель тестирования проекта:

- Проверить работу веб-приложения под определённой нагрузкой
- Проверить корректность результатов
- Проверить разработанное веб-приложение на кроссплатформенность
- Проверить приложение на удобство пользователя при взаимодействии с разными параметрами интерфейса (кнопки, цвета, выравнивание и т. д.).
- Протестировать модели машинного обучения

Итоговыми документами процесса тестирования буду являться:

- тест план,
- отчет о результатах тестирования, включающий в себя описание тестовых сред,
- отчет с написанными тест-кейсами,
- отчет о дефектах.

Тестирование предполагается вести в ручном и автоматизированном, режимах.

Предполагается использование автоматизированных систем тестирования Adaptivator, Browsershots, Loadly, PageSpeed Insights, ValidatorW3C.

2 СТРАТЕГИИ ПРОЦЕССА ТЕСТИРОВАНИЯ

2.1 ВИДЫ ТЕСТИРОВАНИЯ

2.1.1 Тестирование кроссплатформенности

Для проверки адаптивности сайта будет использован сервис Adaptivator <http://adaptivator.ru>. Он проводит общую оценку качеству оптимизации, дает советы по исправлению ошибок. Есть возможность посмотреть, как выглядит сайт на различных популярных устройствах.

Кроссплатформенное тестирование будет включать в себя следующие ОС и браузеры на данных ОС.

Операционная система Linux: Arora 0.11, Chrome 48.0, Dillo 3.0, Eipphany 3.4, Firefox 61.0, Iceweasel 38.0, Konqueror 4.8, Luakit 1.8, Lynx 2.8, Midori 0.5, Opera 12.2, Rekonq 1.1, SeaMonkey 2.7

BrowserStack обеспечивает тестирование веб-браузера в режиме реального времени с мгновенным доступом ко всем настольным и мобильным устройствам. Нет необходимости устанавливать приложение, поскольку оно основано на облаке. Для отладки и кросс-браузерного тестирования используются предустановленные инструменты разработчика.

2.1.2 Тестирование моделей машинного обучения

Тестирование моделей будет проходить на kaggle

2.1.3 Тестирование пользовательского интерфейса

Тестирование пользовательского интерфейса. Графический интерфейс пользователя (Graphical user interface, GUI) –разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений.

Проверяется в целом общий вид приложения и в отдельности формы, расположенные на странице.

Тестирование пользовательского интерфейса. Графический интерфейс пользователя (Graphical user interface, GUI) –разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений.

Проверяется в целом общий вид приложения и в отдельности формы, расположенные на странице.

2.1.4 Тестирование производительности

Тестирование производительности разобьём на несколько тестов

Основные типы тестирования и вопросы, которые они решают представлены в таблице.

№	Вид тестирования	Вид тестирования по английский	Вопрос на который отвечает тестирование
1	Нагрузочное тестирование	Load Testing	Достаточно ли быстро работает система?
2	Тестирование стабильности	Stability Testing	Достаточно ли надёжно работает система на долгом интервале времени?

3	Тестирование отказоустойчивости	Failover Testing	Сможет ли система переместиться сама на другой сервер в случае сбоя основного сервера?
4	Тестирование восстановления	Recovery Testing	Как быстро восстановится система?
5	Стрессовое тестирование	Stress Testing	Что произойдет при незапланированной нагрузке?
6	Тестирование объемов	Volume Testing	Как будет работать система, если объем базы данных увеличится в 100 раз?
7	Тестирование масштабируемости	Scalability Testing	Как будет увеличиться нагрузка на компоненты системы при увеличении числа пользователей?
8	Тестирование потенциальных возможностей	Capacity Testing	Какое количество пользователей может работать?
9	Конфигурационное тестирование	Configuration Testing	Как заставить систему работать быстрее?
10	Тестирование сравнения	Compare Testing	Какое оборудование и ПО выбрать?

В таблице 1 представлены результаты разработки тестов на 17.04.2020 года

Таблица 1. Результаты разработки тестов

Вид тестирования	Количество тестов Запланировано/написано
Тестирование кроссплатформенности	15/0
Тестирование моделей машинного обучения	10/0
Тестирование пользовательского интерфейса	5/0
Тестирование производительности	10/0