Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

*Институт вычислительной математики и информационных технологий*

**ТЕСТ ПЛАН ПО ПРОЕКТУ**

**“Список рекомендаций. Автомобили”**

Составитель Ларичев В. В.

(ФИО)

Руководитель Фаттахов И.Р. (согласовано) 09.05.2024

(ФИО) (подпись)

Дата составления документа

07.05.2024

Казань, 2024 год

Оглавление

[Введение 3](#_heading=h.gjdgxs)

[Объекты тестирования 4](#_heading=h.30j0zll)

[Функции, которые нужно протестировать 5](#_heading=h.1fob9te)

[Функции, которые не нужно тестировать 6](#_heading=h.3znysh7)

[Тестовая документация на проекте 7](#_heading=h.2et92p0)

[Подходы 8](#_heading=h.tyjcwt)

[Критерии прохождения / провала тестирования 9](#_heading=h.3dy6vkm)

[Оценка рисков 10](#_heading=h.1t3h5sf)

[Инструменты для работы 11](#_heading=h.4d34og8)

[Окружение 12](#_heading=h.17dp8vu)

[Сроки 13](#_heading=h.3rdcrjn)

# **Введение**

В рамках этого проекта команда QA приступает к тщательному и всестороннему анализу программного продукта с целью обеспечения его безупречного качества и надежности. Наша задача — провести глубокое функциональное и UI-тестирование, чтобы обнаружить и устранить любые дефекты и несоответствия до того, как продукт будет представлен пользователям. Мы стремимся создать программное обеспечение, которое не только соответствует заданным бизнес-целям, но и превосходит ожидания пользователей, обеспечивая высокий уровень удовлетворенности и эффективности использования.

# **Объекты тестирования**

В рамках тестирования будут охвачены следующие ключевые функциональные аспекты:

* Регистрация и вход в приложение;
* Настройка и сохранение предпочтений пользователя;
* Просмотр и фильтрация объявлений по предпочтениям;
* Добавление объявлений в закладки;
* Скрытие объявлений;
* Пропуск объявлений;
* Просмотр списка понравившихся объявлений;
* Просмотр списка скрытых объявлений;
* Отправка текущей подборки пользователю на почту;
* Вход в приложение через Яндекс;
* Показ общей подборки для нескольких пользователей.

# **Функции, которые нужно протестировать**

1. Регистрация и вход в приложение: Проверка процесса регистрации нового пользователя и входа в существующий аккаунт.
2. Настройка и сохранение предпочтений пользователя: Проверка возможности выбора и сохранения предпочтений пользователя, таких как марка автомобиля, тип кузова и другие параметры.
3. Просмотр и фильтрация объявлений по предпочтениям: Проверка корректности отображения объявлений в соответствии с выбранными предпочтениями пользователя.
4. Добавление объявлений в закладки: Проверка функционала добавления объявлений в список понравившихся.
5. Скрытие объявлений: Проверка функционала скрытия объявлений, чтобы они не отображались в будущем.
6. Пропуск объявлений: Проверка функционала пропуска объявлений, не влияющего на дальнейшую выборку.
7. Просмотр списка понравившихся объявлений: Проверка корректности отображения списка объявлений, добавленных в закладки.
8. Просмотр списка скрытых объявлений: Проверка корректности отображения списка объявлений, скрытых пользователем.
9. Отправка текущей подборки пользователю на почту: Проверка корректности отправления подборки автомобилей пользователю на почту.
10. Вход в приложение через Яндекс: Проверка возможности авторизации в аккаунт через Яндекс.
11. Показ общей подборки для нескольких пользователей: Проверка корректности отображения общей подборки для нескольких пользователей в соответствии с их предпочтениями.

# **Функции, которые не нужно тестировать**

1. Автоматическое обновление списка объявлений без вмешательства пользователя: Этот функционал может быть исключен из тестирования, так как он не влияет на основные функции приложения и может быть автоматически проверен через автоматизированные тесты.
2. Оптимизация производительности приложения: Тестирование оптимизации производительности, такие как загрузка страниц и отзывчивость приложения, может быть автоматизировано и не требует ручного тестирования.
3. Безопасность и защита данных пользователя: Хотя безопасность является критически важной, некоторые аспекты, такие как защита от SQL-инъекций или XSS-атак, могут быть автоматизированы и не требуют ручного тестирования.

# **Тестовая документация на проекте**

Применение чек-листов как ключевого инструмента для организации и проведения тестирования. Эти документы позволяют систематизировать и структурировать процесс тестирования, обеспечивая его эффективность и качество. Тест-кейсы будут разработаны для конкретных модулей и сценариев, где требуется более глубокий анализ, что позволит обеспечить полное покрытие функциональности и удовлетворить все требования к качеству продукта.

Все элементы документации будут регулярно пересматриваться, корректироваться и обновляться в соответствии с изменениями в требованиях проекта. Это обеспечит актуальность и релевантность тестовых случаев, позволяя адаптироваться к новым условиям и требованиям, а также исключать устаревшие или неактуальные тесты. Мы стремимся к тому, чтобы наша документация была ясной, последовательной и легко доступной для всех участников команды, что способствует эффективному сотрудничеству и обмену знаниями внутри команды.

# **Подходы**

В рамках этого проекта используются различные методы тестирования, включая не только ручные виды тестирования, но и модульные тесты:

* Функциональное
* Нефункциональное
* Тестирование локализации;
* Тестирование UI и Удобства использования;

Также планируется использование как позитивных, так и негативных тестов, чтобы убедиться, что продукт может справиться с различными сценариями использования, включая как нормальные, так и нестандартные ситуации.

# **Критерии прохождения / провала тестирования**

Каждый тест-кейс будет обозначен как «Untested» (непроверенный), «Skipped» (пропущен), «Passed» (пройден), «Failed» (провален), «Blocked» (заблокирован) в зависимости от пяти критериев:

1. Наличие и серьезность багов;
2. Уровень успешно выполненных требований;
3. Соответствие стандартам безопасности;
4. Удобство использования;
5. Повторное тестирование после исправления.

# **Оценка рисков**

Риски:

* Отсутствие опыта написания модульных тестов;
* Недостаточное понимание требований заказчика;
* Технические проблемы, связанные с использованием новых технологий.
* Нехватка времени для тестирования всех функций приложения;
* Технические проблемы с оборудованием или программным обеспечением;
* Ошибки в коде, обнаруженные после завершения тестирования;
* Изменения в требованиях заказчика после завершения тестирования;
* Непредвиденные обстоятельства, такие как стихийные бедствия или технические сбои.

Для предотвращения этих рисков рекомендуется:

* Проведение обучения и семинаров для разработчиков и тестировщиков по написанию модульных тестов;
* Проведение детального анализа требований и использование методологий, таких как Agile, для улучшения взаимодействия с заказчиком и более точного понимания его потребностей;
* Проведение обучения команды для работы с новыми технологиями;
* Применение методологии тестирования на основе рисков для определения наиболее критических функций и фокусировки на них, а также использование автоматизированных тестов для ускорения процесса тестирования;
* Использование гибких методологий разработки для быстрого адаптирования к изменениям требований и минимизации затрат на изменения;

# **Инструменты для работы**

* [Google Sheets](https://www.google.com/sheets/about/) - для ведения чек-листов
* [Excel](https://www.microsoft.com/ru-ru/microsoft-365/excel) - для отслеживания и управления багами
* [Чек-лист](https://docs.google.com/spreadsheets/d/1koC1T8Hap9PfGRhdu-UOeLlNwCIba_c8/edit?usp=sharing&ouid=115662442009399807427&rtpof=true&sd=true) - список пунктов, которые нужно проверить
* [Visual Studio](https://visualstudio.microsoft.com/ru/) - для написания unit-тестов

# **Окружение**

Тестовый контур:

NUnit: Один из самых популярных фреймворков для модульного тестирования в .NET. Предоставляет богатые возможности для написания и запуска тестов на C#.

# **Сроки**

06.04.2024 – 11.04.2024

Фаза, в которой разработчиком еще ничего не написано.

Написание тест-плана и чек-листов, тестирование документации и макетов, участие в составлении ТЗ.

11.04.2024 – 18.04.2024

Стадия интенсивной разработки программного кода разработчиком.

18.04.2024 – 20.04.2024

В этой фазе проводится тест ран, чтобы выявить ранние ошибки и проблемы в разработке.

21.04.2024 – 23.04.2024

Написание отчёта об ошибках, создание модульных тестов для существующих методов и ручное тестирование.

24.04.2024 – 25.04.2024

Финальное тестирование и демонстрация проекта заказчику.

06.05.2024 – 10.05.2024

Фаза, в которой разработчиком не реализованы новые функции.

Внесение корректировок в тест-план и чек-лист, тестирование документации и макетов, участие в составлении ТЗ.

10.05.2024 – 14.05.2024

Стадия интенсивной доработки программного код разработчиком.

11.05.2024 – 14.05.2024

Написание отчёта об ошибках, создание модульных тестов для существующих методов и ручное тестирование.

13.05.2024 – 14.05.2024

В этой фазе проводится тест ран, чтобы выявить ранние ошибки и проблемы в разработке.

14.05.2024 – 18.05.2024

Финальное тестирование и демонстрация проекта заказчику.