Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

*Институт вычислительной математики и информационных технологий*

**ТЕСТОВЫЙ ПЛАН**

Проект №2 по дисциплине «Разработка приложений на С#» на тему «Список рекомендаций»

Обучающийся Назипова А. М., 09-322

(подпись)

Заказчик

(подпись)

Руководитель Фаттахов И.Р. (согласовано) 19.04.2024

(подпись

Дата сдачи

Казань, 2024 год

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_heading=h.gjdgxs)

[ОБЪЕКТЫ ТЕСТИРОВАНИЯ 3](#_heading=h.30j0zll)

[ФУНКЦИИ, КОТОРЫЕ НЕОБХОДИМО ПРОТЕСТИРОВАТЬ 3](#_heading=h.1fob9te)

[ФУНКЦИИ, КОТОРЫЕ НЕ НУЖНО ТЕСТИРОВАТЬ 4](#_heading=h.3znysh7)

[ТЕСТОВАЯ ДОКУМЕНТАЦИЯ 4](#_heading=h.2et92p0)

[ПОДХОДЫ 4](#_heading=h.tyjcwt)

[КРИТЕРИИ ПРОХОЖДЕНИЯ/ПРОВАЛА ТЕСТИРОВАНИЯ 4](#_heading=h.3dy6vkm)

[ОЦЕНКА РИСКОВ 6](#_heading=h.1t3h5sf)

[ИНСТРУМЕНТЫ ДЛЯ РАБОТЫ 7](#_heading=h.4d34og8)

[ОКРУЖЕНИЯ 7](#_heading=h.2s8eyo1)

[СРОКИ 8](#_heading=h.17dp8vu)

# ВВЕДЕНИЕ

Наша команда проводит функциональное и UI-тестирование приложения "Список рекомендаций" с целью выявления возможных ошибок. Мы осуществляем детальное тестирование заявленных функциональных возможностей, направленное на помощь в достижении учебных целей продукта.

# ОБЪЕКТЫ ТЕСТИРОВАНИЯ

1. Запуск приложения;
2. Локализация;
3. Личный кабинет;
4. Рекомендации;
5. Подборки;
6. Выставление оценок.

# ФУНКЦИИ, КОТОРЫЕ НЕОБХОДИМО ПРОТЕСТИРОВАТЬ

1. Запуск приложения:

Наличие интерфейса, корректное подключение к базе данных.

1. Локализация:

На странице входа должна быть кнопка для смены языка. При нажатии на кнопку весь текст в приложении переводится с русского языка на английский.

1. Главная страница:

Пользователю выводится список заведений. Кликнув на заведение, пользователь может добавить его в Избранное, поставить ему оценку, а затем нажать на кнопку "Показать рекомендации", после чего открывается окно "Рекомендации".

1. Рекомендации:

На основе понравившихся заведений, который пользователь «лайкнул» на главной странице, в рекомендациях выводятся рестораны/бары. В списке рекомендация можно нажать на любое из заведений, тогда в правой части экрана выводится информация о заведении.

1. Подборки:

Пользователь может просмотреть подборку, встроенную по умолчанию, — «Избранное», или добавить свою.

1. Выставление оценок:

Пользователь может поставить заведению «лайк» или «дизлайк». В соответствии с поставленной оценкой заведение будет выводиться в рекомендациях или наоборот.

# ФУНКЦИИ, КОТОРЫЕ НЕ НУЖНО ТЕСТИРОВАТЬ

Таких функций нет.

# ТЕСТОВАЯ ДОКУМЕНТАЦИЯ

Планируется использование чек-листов в виде основного артефакта тестирования.

Все элементы документации будут пересматриваться, правиться и обновляться по ходу проекта в соответствии с ТЗ.

# ПОДХОДЫ

В данном проекте будут применяться как позитивные, так и негативные тесты. Будут использованы ручные методы тестирования, такие как:

1. Функциональное
2. Нефункциональное

* Тестирование локализации

1. Связанное с изменениями

* Регрессионное тестирование

1. Модульное тестирование

* Unit-тесты

# КРИТЕРИИ ПРОХОЖДЕНИЯ/ПРОВАЛА ТЕСТИРОВАНИЯ

У каждого тест-кейса устанавливается один из следующих статусов: untested, passed, failed, blocked, skipped.

Тест-кейсу присваивается статус Passed, если:

1. Произошло успешное выполнение ожидаемых действий и получение ожидаемого результата в соответствии с заданными шагами и ожидаемым поведением приложения.
2. Отсутствует выявленные ошибки, недочеты или неполадки при прохождении тест-кейса.
3. Полученные результаты соответствуют заданным критериям и требованиям, установленным для данного тест-кейса.
4. Проведение повторного исполнения тест-кейса не приводит к возникновению ошибок или проблем и результат остается совпадающим с ожидаемым.

Тест-кейсу присваивается статус Failed, если:

1. Выполнение ожидаемых действий неудовлетворительное или получен некорректный результат, не соответствующий заданным шагам и ожидаемому поведению приложения.
2. Выявлены ошибки, недочеты или неполадки при прохождении тест-кейса, которые могут повлиять на функциональность или качество приложения.
3. Полученные результаты не соответствуют заданным критериям и требованиям, установленным для данного тест-кейса.
4. Повторное исполнение тест-кейса приводит к аналогичным ошибкам или проблемам, подтверждая неудачный результат.

Тест-кейсу присваивается статус Blocked в случаях:

* + - 1. Недоступности необходимого окружения или ресурсов для выполнения теста.
      2. Неработоспособности или отсутствие необходимого программного обеспечения.
      3. Изменений в функционале проекта, которые могут повлиять на корректность выполнения тест-кейса.
      4. Необходимости дополнительной информации или данных для выполнения теста.

Важно, чтобы статус Blocked был установлен временно, и чтобы проблемы или препятствия были устранены для успешного выполнения тест-кейса в будущем.

Тест-кейсу присваивается статус Skipped, если:

1. Тест-кейс относится к функционалу или сценарию, который временно не активен или находится в процессе разработки.
2. Тест-кейс дублирует или повторяет уже выполненные тесты, и его выполнение лишено смысла.

# ОЦЕНКА РИСКОВ

* + - 1. В ходе работы над проектом выявлены следующие зоны риска:

Работа с базой данных с использованием СУБД SQLite, а также использование Entity Framework в этом проекте командой осуществляется впервые, из-за чего предположительно могут возникнуть задержки на данном этапе работы.

* + - 1. В команде могут возникнуть проблемы с коммуникацией и недопонимания, вследствие чего может пострадать качество исполнения продукта. Например, один участник команды долго не выходит на связь, из-за чего процесс выполнения проекта может задержаться на неопределенный срок.

Способы избежания рисков:

Прочтение и углубленное изучение соответствующей документации. Также необходимо поддерживать связь с заказчиком-разработчиком, который может вносить корректировки в код программы и делать замечания по поводу стиля кода. Для этого обязанности разработчика были поделены на несколько основных задач, после которых предположительно нужно связаться с заказчиком.

Участникам команды необходимо с ответственностью подходить к коммуникации с коллегами в рамках проекта. Тимлид берет на себя задачу обеспечения работоспособной атмосферы в коллективе, в случае необходимости налаживает контакты между участниками. Важной составляющей являются регулярные встречи в онлайн или оффлайн формате, на которых каждый из участников команды может поделиться промежуточными результатами работы.

# ИНСТРУМЕНТЫ ДЛЯ РАБОТЫ

* [Microsoft Excel](https://docs.google.com/spreadsheets/d/1PMWKlkZtD5OrFnSIX1WyiJd0B6gIPs5R/edit?usp=drive_link) – для ведения Чек-листа и Баг репорта

# ОКРУЖЕНИЯ

Тестовый контур.

# СРОКИ

Команда планирует выполнить поставленные задачи в следующие сроки:

6.04 – 7.04

Получение темы от заказчика.

8.04 – 9.04

Обсуждение ключевых аспектов ТЗ и интерфейса всей командой с помощью командного видеозвонка.

9.04 – 12.04

Аналитик составляет Техническое задание и утверждает интерфейс. Написание тест-плана тестировщиком.

13.04 – 18.04

Фаза, в которой разработчик предположительно уже реализовал визуальную часть приложения и подключение к базе данных. Утверждение ТЗ, тест-плана.

18.04 – 21.04

Предполагается, что за этот период разработчик уже написал основную часть функционала. Написание юнит тестов, активное ведение баг-репортов, регрессионное тестирование.

22.04 – 25.04

Финальные проверки. Внесение корректировок в код в соответствии с указанием заказчика-разработчика.

26.04 – 04.05

Подготовка презентации проекта.

04.05

Защита проекта.