**Цель лабораторной работы**

Целью лабораторной работы является изучение и практическое применение методов тестирования программного проекта с целью обеспечения его качества, надежности и соответствия требованиям. В ходе работы необходимо провести анализ выбранного программного проекта и разработать стратегию тестирования, которая позволит выявить потенциальные дефекты и улучшить качество программного продукта.

**Описание программного проекта**

Программный проект представляет собой сервис динамического сегментирования пользователей для компании "Авито". Основной задачей сервиса является централизация работы с проводимыми экспериментами на платформе, такими как тестирование новых продуктов, интерфейсов, скидочных акций и прочих функциональных изменений. В результате архитектурного решения комитета необходимо разработать и реализовать сервис, который позволит управлять сегментами пользователей и определять их участие в различных экспериментах.

Основные сценарии использования сервиса включают в себя создание, изменение и удаление сегментов пользователей, а также добавление и удаление пользователей из сегментов. Для каждого пользователя ведется список сегментов, в которых он состоит. Примерами сегментов могут быть тесты голосовых сообщений, новых услуг продвижения и различные скидочные акции.

Целью проекта является обеспечение удобного и эффективного инструмента для работы с экспериментами на платформе "Авито", что позволит повысить эффективность и результативность проводимых тестов и анализировать влияние изменений на поведение пользователей и показатели платформы.

**Элементы классификации тестирования**

*По уровню тестирования:*

* Модульное тестирование: проверка отдельных компонентов сервиса, таких как методы добавления и удаления сегментов, а также методы работы с пользователями.
* Интеграционное тестирование: проверка взаимодействия между различными модулями сервиса, например, проверка корректной работы API для управления сегментами и пользователями.
* Системное тестирование: проверка работы сервиса в целом, включая его функциональность и соответствие требованиям заказчика.

b. *По методу тестирования:*

* Функциональное тестирование: проверка соответствия функциональных требований, например, проверка корректной работы методов создания и удаления сегментов.
* Нагрузочное тестирование: оценка производительности сервиса при различных нагрузках, например, при одновременном добавлении большого количества пользователей в сегменты.
* Совместимостное тестирование: проверка работы сервиса на различных платформах и в различных окружениях.

c. *По видам тестирования:*

* Позитивное тестирование: проверка корректной работы функционала при правильных входных данных, например, успешное создание сегмента при передаче правильных параметров.
* Негативное тестирование: проверка обработки ошибочных ситуаций и некорректных данных, например, обработка ошибки при попытке удалить несуществующий сегмент.
* Регрессионное тестирование: проверка отсутствия регрессии функционала, например, после внесения изменений в код сервиса.

d. *По уровням тестирования:*

* Unit-тестирование: тестирование отдельных компонентов или модулей сервиса, например, тестирование отдельных методов API.
* Интеграционное тестирование: проверка взаимодействия между различными компонентами сервиса, например, тестирование работы API с базой данных.
* Системное тестирование: проверка работы сервиса в целом, в реальных условиях использования, включая все его компоненты и взаимодействие с другими системами.

**Примеры дефектов**

*Модульное тестирование:*

* Дефект в методе создания сегмента, который приводит к неправильному формированию данных в базе данных.
* Ошибки в логике метода удаления пользователя из сегмента, из-за которых пользователь остается в сегменте после удаления.

b. *Интеграционное тестирование:*

* Неправильная обработка ответа от внешнего сервиса при интеграции с системой аналитики, что приводит к неверному определению сегментов для пользователей.
* Ошибка в механизме синхронизации данных между сервисом и базой данных, вызывающая несоответствие данных.

c. *Системное тестирование:*

* Проблемы с производительностью сервиса при одновременном добавлении большого количества пользователей в сегменты.
* Некорректное отображение данных о сегментах в пользовательском интерфейсе из-за ошибок в логике запросов к API сервиса.

**Выводы**

Проведенный анализ программного проекта и выполненное тестирование позволяют сделать следующие выводы:

1. *Качество и надежность сервиса:* Проведенное тестирование позволяет утверждать, что сервис динамического сегментирования пользователей для компании "Авито" соответствует требованиям качества и надежности. Выявленные дефекты были успешно исправлены, что позволяет гарантировать стабильную работу сервиса.
2. *Эффективность тестирования:* Использование различных методов тестирования, включая модульное, интеграционное и системное тестирование, позволило выявить широкий спектр потенциальных проблем и дефектов. Это дает уверенность в том, что сервис готов к использованию и обладает высоким уровнем качества.
3. *Дальнейшие рекомендации:* Для дальнейшего улучшения качества сервиса рекомендуется регулярное проведение тестирования после внесения изменений и обновлений, а также мониторинг работы сервиса в реальном времени для выявления и устранения любых проблем.

В целом, проведенная работа по тестированию программного проекта позволяет утверждать, что сервис динамического сегментирования пользователей для компании "Авито" соответствует высоким стандартам качества и готов к использованию.

**Список использованных источников**

1. [Github - Задание для стажёра Backend в Avito Tech](https://github.com/avito-tech/backend-trainee-assignment-2023)
2. [Github - Репозиторий Очирова Алдара по управлению качеством программного обеспечения](https://github.com/Aoladiy/software-systems-quality-management)
3. [Тестирование программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)
4. [Статья на Хабре про тестирование документации](https://habr.com/ru/articles/346290/)
5. [Документация по Go](https://go.dev/doc/)