

Заказчик:	НИУ ВШЭ, Международная лаборатория интеллектуальных систем и структурного анализа, Паринов Андрей Андреевич
Название проекта:	Клиент-серверное iOS приложение Ассистента Студента
Исполнители	Зубарева Наталия Дмитриевна, Мостачев Андрей Олегович, Поволоцкий Виктор Александрович, Сальникова Алиса Дмитриевна

1. Описание выбранных методов тестирования

Требования, предъявляемые к тестированию проекта:

- Интерфейс приложения должен быть протестирован на эмуляторе разных IOS-смартфонов для проверки адаптируемости верстки и соответствия содержимого требованиям к интерфейсу.
- Серверная часть, а именно запросы к базе данных, должна быть протестирована на основную логику use-case. Должна быть проверена функциональность всех ссылок, присутствующих как end-point-ы сервера, также должно быть проверено их соответствие требуемым методам для обмена информацией с сервером.
- Должно быть проведено тестирование use-case-ов пользователя в приложении для удостоверения, что все функционирует в соответствии с требованиями, и все необходимые возможности присутствуют.
- Сетевой слой клиентской части приложения (взаимодействие с сервером через запросы) должен быть протестирован в целях выявления багов и обеспечения надежности.
- Должно быть проведено тестирование-оценка модели машинного обучения на основании тегов с целью проверки ее относительной предиктивной способности.

Методы, выбранные для реализации тестирования:

- **ручное тестирование** будет применено для тестирования:
 - **интерфейса**, так как его имеет смысл проверить визуально. Ручной режим тестирования обоснован большей легкостью и доступностью по сравнению с автоматизированным тестированием. Кроме того, так будет протестировано удобство использования, оцениваемое субъективно тестировщиком, что невозможно при автоматизированном тестировании.
 - **контрактов API запросов к серверу**, в силу специфики организации сервера. Так как сервер доступен только в режиме удаленного устройства и обращения через командную строку, настройка автоматических тестов для него была бы более времязатратной, чем проведение ручного тестирования, при том что их эффективность была бы сравнима. Так как запросы передаются в читаемом формате, тестировщик может оценить их соответствие ожидаемому результату и, в случае различий между ожидаемым форматом и результатом запроса, адаптировать либо серверную часть, либо клиентскую, в соответствии с своими соображениями.
 - **нейросетевой модели**, так как модель, будучи нейросетью, обучающейся без учителя, не облагается строгими, например численными, требованиями, которые можно было бы проверить автоматически. В рамках визуальной

оценки данных и результатов работы модели будет установлено, выходит ли модель за границы осмысленных значений рекомендаций, и насколько она в целом логична. Также важную роль играют данные, на которых модель будет тренироваться и работать: требуется сделать поправку на их сбалансированность, осмысленность и т.д. Так как от модели требуется, в общем смысле, автоматизировать ментальный процесс подбора рекомендаций, который обычно осуществляется людьми, измерение качества модели путем человеческой оценки кажется наиболее целесообразной.

- **системное ручное тестирование** будет применено для тестирования **пользовательских историй**, так как для этого вида тестирования имеет наибольший смысл именно ручное тестирование. Требуется не только проверить реализуемость всех действий, но и понятность интерфейса, его удобство и проанализировать необходимость и возможные пути их улучшения.
- **автоматизированное модульное тестирование** будет применено для тестирования **сетевого слоя**, так как реализовать модульные тесты запросов к серверу в проекте клиентского приложения сравнительно несложно, и это поддержит ручное тестирование запросов. Также это обеспечит дальнейшую поддерживаемость кода клиентской части, которая более вероятно чем серверная часть будет подвергнута расширению и изменениям.
- другие виды тестирования не будут осуществлены отдельно, могут быть затронуты в рамках вышеописанных методов. В частности, нагрузочное тестирование не будет проводиться, так как нагрузка на сервер не предполагается быть больше, чем обеспечиваемая сервером выбранного типа по умолчанию. Тестирование безопасности будет обеспечено в рамках проверки доступа к серверу (с помощью тестов попыток неавторизованного доступа к данным) и пользовательских историй.

2. Баг-репорт

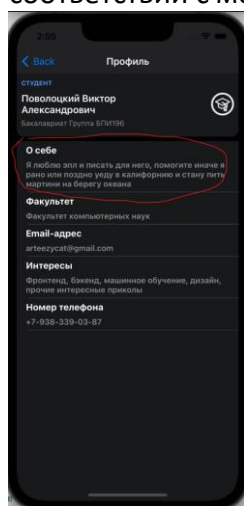
1. **Заголовок:** Список тэгов по проекту не фильтруется по его идентификатору при отправке запроса `requirements_by_project/id`
 - **Подробное описание:** При попытке получить агрегированные тэги конкретного проекта по его идентификатору не происходит фильтрации агрегированных данных тэгов. Отображаются тэги для всех проектов, а не только для нужного.
 - **Откуда:** ручное тестирование запросов к серверу
 - **Шаги для воспроизведения:**
 - получить идентификатор проекта, по которому требуется посмотреть тэги, - id, например 5
 - ввести в браузере `http://84.201.135.211:8000/requirements_by_project/id/`
 - **Результат:** тэги, относящиеся ко всем проектам [{"project_id": 4,...}, {"project_id": 8,...}...]
 - **Ожидаемый результат:** тэги, относящиеся только к проекту id [{"project_id": 5,...}, {"project_id": 5,...}...]
 - **Критичность:** 1/5
 - **Приоритет:** 1/5
 - **Устранено:** частично. После нескольких безуспешных попыток принято решение не исправлять это на стороне сервера, а осуществлять фильтрацию на стороне клиента.
2. **Заголовок:** Список заявок на проекты от студента не фильтруется по его идентификатору при отправке запроса `applications_by_student/id`
 - **Подробное описание:** При попытке получить агрегированные заявки конкретного студента по его идентификатору не происходит фильтрации агрегированных данных. Отображаются заявки всех студентов, а не только для нужного.
 - **Откуда:** ручное тестирование запросов к серверу
 - **Шаги для воспроизведения:**
 - получить идентификатор студента, по которому требуется посмотреть заявки, - id, например 5
 - ввести в браузере `http://84.201.135.211:8000/applications_by_student/id/`
 - **Результат:** заявки всех студентов [{"student_id": 4,...}, {"student_id": 8,...}...]
 - **Ожидаемый результат:** заявки студента с id [{"student_id": 5,...}, {"student_id": 5,...}...]
 - **Критичность:** 1/5
 - **Приоритет:** 1/5
 - **Устранено:** частично. После нескольких безуспешных попыток принято решение не исправлять это на стороне сервера, а осуществлять фильтрацию на стороне клиента.
3. **Заголовок:** Список информации студента не фильтруется по его идентификатору при отправке запроса `information_by_student/id`
 - **Подробное описание:** При попытке получить подробную информацию конкретного студента по его идентификатору не происходит фильтрации данных. Отображается информация всех студентов, а не только нужного.
 - **Откуда:** ручное тестирование запросов к серверу
 - **Шаги для воспроизведения:**

- получить идентификатор студента, для которого требуется посмотреть информацию, - id, например 5
 - ввести в браузере http://84.201.135.211:8000/information_by_student/id/
 - **Результат:** информация всех студентов [{"id": 4,...}, {"id": 8,...}...]
 - **Ожидаемый результат:** информация студента с id [{"id": 5,...}, {"id": 5,...}...]
 - **Критичность:** 1/5
 - **Приоритет:** 1/5
 - **Устранено:** частично. После нескольких безуспешных попыток принято решение не исправлять это на стороне сервера, а осуществлять фильтрацию на стороне клиента.
- 4. Заголовок:** Список интересов студента не фильтруется по его идентификатору при отправке запроса [interests_by_student/id](http://84.201.135.211:8000/interests_by_student/id/)
- **Подробное описание:** При попытке получить агрегированные тэги интересов конкретного студента по его идентификатору не происходит фильтрации агрегированных данных тэгов. Отображаются тэги для всех студентов, а не только для нужного.
 - **Откуда:** ручное тестирование запросов к серверу
 - **Шаги для воспроизведения:**
 - получить идентификатор студента, по которому требуется посмотреть тэги, - id, например 5
 - ввести в браузере http://84.201.135.211:8000/interests_by_student/id/
 - **Результат:** тэги интересов, относящиеся ко всем студентам [{"id": 4,...}, {"id": 8,...}...]
 - **Ожидаемый результат:** тэги интересов, относящиеся только к студенту с id [{"id": 5,...}, {"id": 5,...}...]
 - **Критичность:** 1/5
 - **Приоритет:** 1/5
 - **Устранено:** частично. После нескольких безуспешных попыток принято решение не исправлять это на стороне сервера, а осуществлять фильтрацию на стороне клиента.
- 5. Заголовок:** IntegrityError при попытке создания студента через запрос `student_create`
- **Подробное описание:** При попытке создать нового студента по запросу `student_create` с указанием любого email-адреса происходит ошибка "insert or update on table "student" violates foreign key constraint "student__fk" DETAIL: Key (id)=(3) is not present in table "auth_user"."
 - **Причина:** создание студента присоединено к аутентификации пользователя, поэтому таблица ожидает ключ из таблицы пользователей.
 - **Откуда:** ручное тестирование запросов к серверу
 - **Шаги для воспроизведения:**
 - ввести в браузере http://84.201.135.211:8000/student_create
 - заполнить поле email любым значением
 - нажать кнопку post чтобы отправить запрос
 - **Критичность:** 1/5
 - **Приоритет:** 1/5
 - **Устранено:** да. Так как создание студента перенаправлено в аутентификацию пользователя, принято решение удалить этот url из возможных запросов.

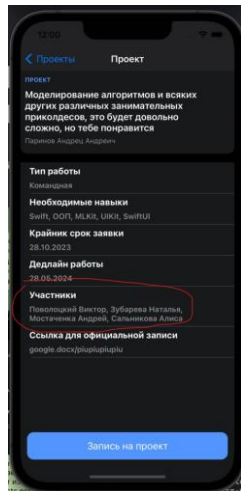
6. **Заголовок:** модель машинного обучения генерирует рекомендации с очень слабо тематически связанными тегами.
- **Подробное описание:** модель иногда выдает результаты с довольно далекими друг от друга тематиками, например “мобильная разработка” и “ассемблер”, при том что корреляция этих тэгов задана как низкая. Возможно, дело в величине популярности тэгов.
 - **Откуда:** визуальная оценка качества работы модели
 - **Шаги для воспроизведения:** посмотреть набор сгенерированных рекомендаций.
 - **Критичность:** 2/5
 - **Приоритет:** 2/5
 - **Устранено:** да. Популярность отдельных тэгов была нормирована, теперь вероятность получения пары не смещается из-за них.

7. **Заголовок:** лишнее поле “о себе” в интерфейсе на странице профиля.

- **Подробное описание:** при открытии в приложении страницы профиля студента можно видеть поле с данными “о себе”, его не должно быть в соответствии с моделями данных.



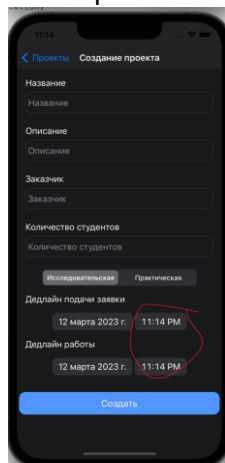
- **Откуда:** ручное визуальное тестирование интерфейса
 - **Шаги для воспроизведения:**
 - открыть приложение
 - сделать логин пользователя
 - открыть экран профиля
 - **Критичность:** 1/5
 - **Приоритет:** 2/5
 - **Устранено:** да. Поле убрано
8. **Заголовок:** неправильный формат поля участников на экране проекта в приложении.
- **Подробное описание:** поле участников должно быть числовым и показывать число участников, которые могут участвовать в проекте. В данный момент поле показывает информацию из мокап данных и имеет неправильный тип.



- **Откуда:** ручное визуальное тестирование интерфейса
- **Шаги для воспроизведения:**
 - открыть приложение
 - сделать логин пользователя
 - открыть экран проектов
 - открыть экран любого проекта нажав на любой проект
- **Критичность:** 1/5
- **Приоритет:** 2/5
- **Устранено:** да. Поле изменено

9. Заголовок: лишнее поле времени в дедлайнах заявки и выполнения проекта на экране создания проекта в приложении

- **Подробное описание:** сейчас на экране присутствуют поля для выбора времени дедлайна, хотя в модели данных такого нет, эта информация никуда не сохраняется.



- **Откуда:** ручное визуальное тестирование интерфейса
- **Шаги для воспроизведения:**
 - открыть приложение
 - сделать логин пользователя
 - открыть экран проектов
 - открыть экран создания нового проекта нажав на плюсикик вверху
- **Критичность:** 1/5
- **Приоритет:** 1/5
- **Устранено:** да. Поле убрано

3. Приёмочное тестирование

Заказчик подтверждает полученные результаты:

№ п/п	Наименование работ в соответствии с ТЗ	Статус (статус заполняет заказчик): выполнено/частично выполнено/не выполнено
1.	Авторизация пользователя в приложении по данным логина и пароля через систему ЕЛК ВШЭ.	ЧАСТИЧНО
2.	Просмотр студентом списка проектов.	ДА
3.	Просмотр студентом подробной информации о проекте.	ДА
4.	Заполнение студентом персональных данных и интересов.	ДА
5.	Подача студентом заявки на участие в проекте и отмена этой заявки.	ДА
6.	Сохранение данных проектов и студентов, синхронизация данных между базой данных и клиентским приложением.	ДА
7.	Предоставление рекомендаций проектов студенту на основании интересов студента и тегов проектов.	ЧАСТИЧНО
8.	Создание собственного инициативного проекта студентом.	ДА
9.	Предоставление возможности электронным сервисам ВШЭ получать доступ к данным приложения.	ДА
10.	Система соответствует прочим нефункциональным требованиям.	ДА
Перечень НЕ выполненных работ (заполняет заказчик):		1. Подключение рекомендательной системы для получения рекомендаций из приложения

Работы, не выполненные на момент приемки, подлежат выполнению в ближайшем релизе до финального отчета, защиты проекта и передачи программы.

Комментарии заказчика к результатам:

Большая часть функционала реализована и протестирована. Доработки к проекту указаны выше.

Заказчик

Личная подпись /Паринов А.А

Ответственный по проекту

Личная подпись _____/Зубарева Н.Д.

Дата: 15.03.2023

