ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разработка цифровой платформы поиска семестрового обмена студентов

Проект: Дочерний сервис Сбербанка

Версия: 1.1

Дата: 16 апреля 2025 г.

Разработчики:
Литовченко Виктория Вадимовна
Руденко Евгения Кирилловна
Леонтьева Анастасия Александровна
Огаркова Анастасия Владимировна
Табунов Иван Дмитриевич

Заказчик: Сбербанк

Контактная информация

Email: simpatulka228@gmail.com

Телеграмм: @vkpnmrv

Оглавление

1. Введение и цели проекта

1.1 Описание проекта

Создание единой цифровой платформы для поиска, подачи заявок и взаимодействия между студентами и вузами-партнерами в рамках программ семестрового академического обмена. Платформа должна решать проблему отсутствия централизованного ресурса с актуальной информацией, коммуникацией и рецензированием программ.

1.2 Цели

- Обеспечить студентов и вузы удобным инструментом для поиска и управления программами обмена.
- Автоматизировать процессы подачи и обработки заявок.
- Создать прозрачную систему отзывов и рейтингов вузов и программ.
- Внедрить AI-модули для персонализированного подбора программ и аналитики.
- Обеспечить высокий уровень безопасности и соответствие требованиям законодательства РФ.

1.3 Целевая аудитория

- Студенты российских и зарубежных вузов, заинтересованные в академическом обмене.
- Представители вузов-партнеров, курирующие программы обмена.
- Администраторы платформы.

2. Общие требования

2.1 Стиль и бренд

- Использовать фирменные цвета Сбербанка: основной зелёный (#1A9F53), синий (#004F95), акцентный фиолетовый (#8549FF).
- Дизайн должен быть современным, минималистичным, адаптивным (mobile-first).
- Использовать фирменный шрифт «SBSans» или аналогичный.

2.2 Технологии

Компонент	Технологии / Описание	Взаимодействие
Frontend	React + TypeScript, Bootstrap (для адаптивности)	Общается с Backend API через HTTP(S)
Backend API	Spring Boot + AI интеграция	Принимает запросы от Frontend, взаимодействует с БД и AI модулем
Database	PostgreSQL	Хранит данные, доступен через Backend API
Chat Server	(Отдельный сервер для чата)	Связан с Frontend и AI модулем
AI Module	GigaChat (модуль ИИ)	Обрабатывает запросы от Backend API и Chat Server
Storage	Документы, медиа	Хранение файлов, доступ через Backend API

Frontend: React + TypeScript, Material UI, Redux, Axios, Bootstrap.

Backend: Java (Spring Boot) + Python (Django/Flask) для AI, Node.js (Express.js) для API.

База данных: PostgreSQL с репликацией.

CI/CD: Docker, Kubernetes, Nginx.

Тестирование: Jest, Pytest, Selenium.

AI: интеграция с GigaChat для персонализации и аналитики.

2.3 Безопасность

- Хранение данных с шифрованием AES-256.
- Соответствие ФЗ-152 (персональные данные).
- Двухфакторная аутентификация (2FA) через SMS или приложение Сбербанк Онлайн.
- Защита от спама (капча, лимиты на сообщения).
- Регулярные аудиты безопасности.

2.4 Языковая поддержка

- Русский и английский в первой версии.
- Возможность масштабирования на другие языки (китайский, испанский).

3. Пользовательские сценарии (User Flows)

3.1 Регистрация студента

1. Пользователь заходит на главную страницу → нажимает «Регистрация».
2. Выбирает тип аккаунта «Студент».
3. Заполняет форму: ФИО, email, телефон, вуз, курс, GPA, языковые сертификаты, загружает паспорт.
4. Подтверждает email через ссылку.
5. При первом входе система предлагает заполнить профиль и загрузить документы.
6. Получает приветственное уведомление и рекомендации программ.

3.2 Поиск и подача заявки на программу

1. Студент переходит в каталог вузов.
2. Использует фильтры (страна, специальность, стоимость).
3. Просматривает карточку университета → выбирает программу обмена.
4. Нажимает «Подать заявку».
5. Заполняет заявку, прикрепляет необходимые документы.
6. Отправляет заявку, получает уведомление о статусе «На рассмотрении».

3.3 Управление программами вузом

1. Представитель ВУЗа входит в личный кабинет.
2. Добавляет новую программу обмена с описанием, требованиями, сроками.
3. Просматривает список заявок, фильтрует по статусам.
4. Одобряет или отклоняет заявки, отправляет комментарии студентам.
5. Отвечает на отзывы и сообщения в чатах.

3.4 Оставление отзыва студентом

1. После участия в программе студент получает уведомление с просьбой оставить отзыв.
2. Переходит в раздел отзывов, выбирает программу.

3. Оценивает параметры (учеба, жилье, преподаватели) по шкале 1-5.
4. Пишет текстовый отзыв, прикрепляет фото/видео (опционально).
5. Отправляет отзыв на модерацию.

4. Структура и функционал платформы

4.1 Главная страница

- Логотип и брендовая шапка с кнопками «Регистрация» и «Вход» (правый верхний угол).
- Краткое описание платформы с кнопкой «Узнать больше» (центр экрана).
- Блок «Популярные ВУЗы» — карточки с кнопками «Подробнее».
- Секция «Отзывы студентов» — карусель с возможностью «Читать все отзывы».
- Кнопка чат-бота в нижнем правом углу (закрепленная, круглая).

4.2 Регистрация и вход

- Форма выбора типа аккаунта: «Студент» или «ВУЗ».
- Обязательные поля для студентов: ФИО, email, телефон, вуз, курс, GPA, языковые сертификаты, загрузка паспорта.
- Обязательные поля для вузов: название, страна, контакты куратора, лицензия, описание программ.
- Кнопки: «Зарегистрироваться» (основная), «Войти», «Забыли пароль?».
- Валидация данных и подтверждение email.

4.3 Личный кабинет студента

- Профиль: аватар, ФИО, страна, университет, специальность, курс, уровень владения языками, рейтинг активности.
- Раздел «Мои заявки»: список заявок с фильтрами по статусам:
    - Черновик
    - На модерации
    - Требуются документы
    - Одобрено
    - Отклонено
- Кнопки: «Посмотреть детали», «Отменить заявку».
- Раздел «Рекомендованные программы» — AI-подборка с фильтрами (страна, срок, стоимость).
- Чаты: с представителями вузов, кураторами, студентами-участниками (текст, файлы, видео).
- Раздел «Документы»: загрузка и статус верификации (одобрено/на проверке/отклонено).

- Раздел «Избранные университеты» с возможностью удаления.
- Раздел «Отзывы и опыт»: история участия, добавление новых отзывов.
- Уведомления: новые сообщения, статус заявок, напоминания о дедлайнах.

4.4 Личный кабинет представителя ВУЗа

- Профиль: логотип, название, рейтинг, страна, город, контакты куратора.
- Управление программами: добавление, редактирование, закрытие.
- Список заявок с фильтрами (новые, в процессе, одобрено, отклонено).
- Кнопки действий по заявкам: «Одобрить», «Отклонить», «Связаться».
- Чаты: личные и групповые с кандидатами.
- Раздел «Документы»: просмотр, верификация, запрос дополнительных документов.
- Раздел «Отзывы и рейтинг»: ответы на отзывы.
- Статистика и аналитика: заявки, рейтинги, популярность направлений, экспорт отчетов.
- Уведомления: новые заявки, вопросы, дедлайны.

4.5 Каталог ВУЗов и программ обмена

- Фильтры: страна, город, специальность, длительность, требования к языку, стоимость, дедлайн.
- Карточки вузов: логотип, название, локация, рейтинг, количество программ, язык, стоимость, кнопка «Подробнее».
- Страница университета: фото, описание, карта, год основания, рейтинги (QS, Times, внутренний), количество иностранных студентов.
- Список программ обмена с подробностями и кнопкой «Подать заявку».
- Блок «Популярные ВУЗы» с топ-5 и интерактивной картой.
- Возможность добавлять в избранное.

4.6 Страница программы обмена

- Название, длительность, язык, стоимость, список документов.
- Кнопки: «Подать заявку», «Добавить в избранное».
- Секция отзывов с возможностью оставить отзыв, оценить параметры (учеба, жилье, преподаватели).

- Кнопка «Пожаловаться» на отзывы.

4.7 Система отзывов и рецензирования

- Оставление отзывов с рейтингами по параметрам.
- Модерация: автоматический фильтр спама, ручная проверка.
- Возможность жалоб на отзывы.
- Ответы представителей вузов.

4.8 Чаты и форумы

- Создание тем, ответы, вложения (файлы, изображения, видео).
- Модерация сообщений (скрытая кнопка «Сообщить о нарушении»).
- Уведомления о новых сообщениях.

4.9 Административная панель

- Управление пользователями (блокировка, разблокировка).
- Редактирование контента.
- Настройки AI (параметры рекомендаций).
- Генерация отчетов, сравнение вузов.

4.10 AI-аналитика и отчетность

- Персонализированные рекомендации программ.
- Аналитика по заявкам, популярности вузов и программ.
- Настраиваемые отчеты с экспортом.

5. Требования к безопасности, обработке данных, производительности и нагрузке

5.1 Безопасность данных и пользователей

Шифрование данных
Все персональные и конфиденциальные данные пользователей (студентов, представителей вузов) должны храниться и передаваться с использованием современных криптографических алгоритмов. Для хранения применяется шифрование на уровне базы данных по стандарту AES-256. Передача данных между клиентом и сервером осуществляется по протоколу HTTPS с использованием SSL/TLS сертификатов, обеспечивающих защиту от перехвата и MITM-атак.

Соответствие законодательству
Серверы и инфраструктура размещаются на территории Российской Федерации, что обеспечивает полное соответствие требованиям Федерального закона №152-ФЗ «О персональных данных». Обработка персональных данных осуществляется с соблюдением принципов конфиденциальности и ограниченного доступа.

Аутентификация и авторизация
Для входа в систему реализуется двухфакторная аутентификация (2FA), которая может быть выполнена через SMS-код или интеграцию с приложением Сбербанк Онлайн. Используются современные протоколы аутентификации (OAuth 2.0, OpenID Connect) для безопасного управления сессиями пользователей.
Система реализует строгую разграничительную политику доступа, основанную на ролях (студент, представитель вуза, администратор), с применением принципа наименьших привилегий.

Защита от атак
Веб-приложение должно быть защищено от распространенных уязвимостей:

SQL-инъекции — посредством параметризованных запросов и ORM.

Межсайтовый скриптинг (XSS) — через экранирование пользовательского ввода, Content-Security-Policy (CSP) и HTTP-заголовки (X-XSS-Protection).

Межсайтовая подделка запросов (CSRF) — с использованием токенов CSRF и механизмов двойной проверки (Double Submit Cookies).

Защита от brute-force атак — ограничение количества попыток входа, капча при регистрации и входе.

Мониторинг и аудит безопасности
Внедрены системы мониторинга и логирования безопасности, отслеживающие аномалии в поведении пользователей, попытки несанкционированного доступа и другие подозрительные события. Проводятся регулярные аудиты безопасности и пентесты для своевременного выявления и устранения уязвимостей.

Обновления и патчи
Все компоненты системы (операционная система, веб-сервер, базы данных, используемые библиотеки и фреймворки) регулярно обновляются до последних стабильных версий с целью устранения известных уязвимостей и повышения общей безопасности.

Обучение персонала
Разработчики и администраторы проходят регулярное обучение по современным методам обеспечения безопасности веб-приложений и защите персональных данных.

5.2 Обработка и хранение данных

Персональные данные хранятся в зашифрованном виде на серверах, расположенных в дата-центрах на территории РФ.

Доступ к данным ограничен и контролируется системой прав доступа.

Все операции с персональными данными фиксируются в журнале аудита.

Хранение и обработка данных соответствуют требованиям ФЗ-152 и внутренним политикам безопасности Сбербанка.

Резервное копирование данных выполняется ежедневно с хранением копий не менее 30 дней для восстановления в случае сбоев.

5.3 Производительность и нагрузка

Время отклика
Среднее время отклика сервера не должно превышать 500 миллисекунд при средней и пиковых нагрузках, обеспечивая комфортный пользовательский опыт.

Масштабируемость
Платформа должна быть построена с использованием контейнеризации (Docker) и оркестрации (Kubernetes), что обеспечивает горизонтальное масштабирование и высокую доступность сервиса при росте числа пользователей.

Поддержка нагрузки
Система должна выдерживать одновременную работу до 50 000

пользователей без деградации производительности. Для этого применяется балансировка нагрузки, кэширование и оптимизация запросов к базе данных.

Резервное копирование и восстановление
Ежедневное автоматическое резервное копирование всех данных с возможностью быстрого восстановления. План аварийного восстановления (Disaster Recovery Plan) должен быть разработан и протестирован.

5.4 Рекомендации по реализации

Использовать Web Application Firewall (WAF) для дополнительной защиты от внешних угроз и фильтрации трафика.

Внедрить системы мониторинга активности и аномалий с автоматическим оповещением ответственных лиц.

Применять принципы безопасности на всех этапах разработки (Security by Design).

Регулярно проводить тестирование безопасности (включая пентесты) и обновлять политики безопасности.

6. Тестирование

6.1 Общие положения

Тестирование является неотъемлемой частью разработки и обеспечивает качество, надежность и безопасность платформы. В рамках проекта предусмотрены различные виды тестирования, каждый из которых выполняется на определенных этапах и с использованием соответствующих инструментов.

6.2 Виды тестирования

6.2.1 Модульное тестирование (Unit Testing)

- Цель: Проверка отдельных компонентов системы (функций, методов, классов) на корректность.
- Инструменты: Jest (для JavaScript/TypeScript), Pytest (для Python), JUnit (для Java).
- Описание:
    - Написание тестов для функций и методов, например, для API-обработчиков, бизнес-логики, утилит.
    - Использование моков и фикстур для изоляции тестируемых компонентов.
    - Примеры: тестирование функции обработки заявки, фильтров поиска, расчетных модулей.
- Критерии успешности: 100% покрытие ключевых функций, отсутствие ошибок при автоматическом запуске.

6.2.2 Интеграционное тестирование

- Цель: Проверка взаимодействия между модулями и компонентами системы.
- Инструменты: Postman, Insomnia, встроенные тесты в CI/CD.
- Описание:
    - Тестирование API-эндпоинтов, взаимодействия фронтенда с бекендом.
    - Проверка корректности обмена данными, обработки ошибок.
    - Примеры: тестирование сценариев подачи заявки, получения списка вузов, авторизации.

- Критерии успешности: корректное выполнение сценариев, соответствие спецификациям.

6.2.3 UI/UX тестирование

- Цель: Проверка удобства использования, визуальной согласованности интерфейса.
- Инструменты: Selenium, Cypress.
- Описание:
    - Автоматизированное тестирование сценариев взаимодействия пользователя с интерфейсом.
    - Проверка адаптивности, корректности отображения элементов.
    - Примеры: проверка открытия формы заявки, корректность работы фильтров.
- Критерии успешности: отсутствие ошибок отображения, удобство навигации.

6.2.4 Нагрузочное тестирование

- Цель: Определение устойчивости системы под высокой нагрузкой.
- Инструменты: JMeter, Gatling.
- Описание:
    - Моделирование одновременных пользователей (до 50 000).
    - Проверка времени отклика, стабильности работы при пиковых нагрузках.
    - Тестирование сценариев массовых запросов, регистрации, подачи заявок.
- Критерии успешности: время отклика не превышает 500 мс, время безотказной работы — не менее 99.9%.

6.2.5 Безопасное тестирование (Pentest)

- Цель: Обнаружение уязвимостей, защита данных.
- Инструменты: OWASP ZAP, Burp Suite, собственные сценарии.
- Описание:
    - Проверка защиты от SQL-инъекций, XSS, CSRF, атак на авторизацию.
    - Анализ уязвимостей API, интерфейса.

    - Проведение тестов на проникновение специалистами.
- Критерии успешности: отсутствие критичных уязвимостей, соответствие стандартам безопасности.

6.3 Этапы тестирования

Этап	Описание	Сроки	Ответственный
Подготовка тестовой среды	Развертывание тестовых стендов, подготовка данных	1 неделя	Техническая команда
Модульное тестирование	Автоматические тесты компонентов	На этапе разработки	Разработчики
Интеграционное тестирование	Проверка взаимодействия модулей	После завершения модулей	QA-инженеры
UI/UX тестирование	Автоматизированное и ручное тестирование интерфейса	Перед релизом	QA-инженеры, дизайнеры
Нагрузочное тестирование	Проверка системы под нагрузкой	За 2 недели до релиза	Специалисты по тестам
Безопасное тестирование	Пентесты, анализ уязвимостей	Перед запуском	Специалисты по безопасности

6.4 Критерии качества и успешности тестирования

- 100% покрытие критичных функций модульных тестов.
- Отсутствие критичных уязвимостей по результатам пентестов.
- Время отклика системы — не более 500 мс при пиковых нагрузках.
- Проход всех сценариев интеграционного и UI тестирования без ошибок.
- Минимум ошибок, выявленных на этапе пользовательского тестирования.

6.5 Документация и отчеты

- Ведение журнала тестирования, фиксация ошибок и их статусов.
- Итоговые отчеты по каждому виду тестирования с рекомендациями.
- Акт приемки системы после успешного прохождения всех тестов.

7. План разработки и этапы

Этап	**Описание**	**Сроки**
Аналитика и дизайн	Сбор требований, прототипирование	2 недели
Разработка MVP	Базовый функционал, регистрация, каталог	6 недель
Интеграция AI	Подбор и аналитика программ	2 недели
Тестирование	Функциональное и нагрузочное	2 недели
Запуск и поддержка	Развертывание, исправление багов	Постоянно