

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы Разработка платформы психологического тестирования для корпоративных клиентов
--

УДК 004.774:004.455.1:159.9.072

Студенты

Группа	ФИО	Подпись	Дата
8К61	Балыков Николай Ленскиеич		
8К61	Рыбаченко Иван Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Аксёнов Сергей Владимирович	к.т.н		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСТН ШБИП ТПУ	Спицына Любовь Юрьевна	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП ТПУ	Белоенко Елена Владимировна	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	к.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критерии АИОР
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.	Требования ФГОС (ОПК-1,2,3, ПК-4, 5, 6), критерий 5 АИОР (п. 1.1).
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.	Требования ФГОС (ОПК3, 4, ПК-1, 2, 9), критерий 5 АИОР (п.1.1, 1.2).
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.	Требования ФГОС (ОК-1, 6, ПК-2, 4, 5), критерий 5 АИОР (п. 1.2).
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т.п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.	Требования ФГОС (ОПК2, 3, ПК-3, 4, 5), критерий 5 АИОР (п. 1.3).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретацию полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.	Требования ФГОС (ОПК4, ПК-6, 7), критерий 5 АИОР (п.1.4)
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.	Требования ФГОС (ОПК3, ПК-7, 8, 9), критерий 5 АИОР (п. 1.5)

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критерии АИОР
Универсальные компетенции		
Р7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.	Требования ФГОС (ОКП1, 4, ПК-1, 6, 7), критерий 5 АИОР (п. 2.1).
Р8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности	Требования ФГОС (ОК-5), критерий 5 АИОР (п. 2.2).
Р9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.	Требования ФГОС (ОК6), критерий 5 АИОР (п. 2.3, 2.4)
Р10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.	Требования ФГОС (ОК-2, 3, 4), критерий 5 АИОР (п. 2.5)
Р11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.	Требования ФГОС (ОК7), критерий 5 АИОР (п. 2.6).

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки (специальность): 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ 16.06.20 Чердынцев Е.С.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студентам:

Группа	ФИО
8К61	Балыкову Николаю Ленскиевичу
8К61	Рыбаченко Ивану Александровичу

Тема работы:

Разработка платформы психологического тестирования для корпоративных клиентов	
Утверждена приказом директора (дата, номер)	№ 59-51/с от 28.02.2020

Срок сдачи студентом выполненной работы:	11.06.2020
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	<ol style="list-style-type: none"> 1. Требования к бюджету проекта и его конкурентоспособности 2. Требования к модулю по формированию плана-графика проекта.
---------------------------------	--

Перечень подлежащих исследованию, проектированию и разработке вопросов	<ol style="list-style-type: none"> 1. Обзор предметной области; 2. Проектирование веб-приложения; 3. Программная реализация веб-приложения; 4. Анализ результатов разработки; 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 6. Социальная ответственность
Перечень графического материала	<ol style="list-style-type: none"> 1. Диаграмма вариантов использования 2. Диаграммы в нотации IDEF0; 3. Диаграмма в нотации BPMN; 4. Диаграмма в нотации EPC; 5. Диаграмма в нотации IDEF1X; 6. Пояснительные скриншоты веб-приложения; 7. Матрица SWOT-анализа; 8. Диаграмма Ганта.

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Спицына Любовь Юрьевна
Социальная ответственность	Белоенко Елена Владимировна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.03.2020
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Аксёнов Сергей Владимирович	к. т. н.		

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8К61	Балыков Николай Ленскиевич		
8К61	Рыбаченко Иван Александрович		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки (специальность): Программная инженерия
 Уровень образования Бакалавр
 Отделение школы (НОЦ): Отделение информационных технологий
 Период выполнения: осенний / весенний семестр 2019/2020 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	11.06.2020
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
03.05.2020	Глава 1. Обзор предметной области	10
10.05.2020	Глава 2. Проектирование платформы психологического тестирования для корпоративных клиентов	15
17.05.2020	Глава 3. Программная реализация платформы психологического тестирования для корпоративных клиентов	20
23.05.2020	Глава 4. Анализ результатов разработки платформы психологического тестирования для корпоративных клиентов	15
11.05.2020	Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
24.05.2020	Глава 6. Социальная ответственность	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Аксёнов Сергей Владимирович	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студентам:

Группа	ФИО
8К61	Балыкову Николаю Ленскиевичу
8К61	Рыбаченко Ивану Александровичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	бакалавриат	Направление/специальность	Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Бюджет проекта	Не более 1 миллиона рублей
2. Значение показателя интегральной ресурсоэффективности	Не менее 60 баллов из 100

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Описание потенциальных потребителей; Анализ технических конкурентных решений; SWOT-анализ.
2. Планирование и формирование бюджета научных исследований	Структура работ в рамках научного исследования Определение трудоемкости выполнения работ и разработка графика проведения научного исследования Бюджет проекта
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определение интегрального финансового показателя разработки Определение интегрального показателя ресурсоэффективности разработки Определение интегрального показателя эффективности

Перечень графического материала:

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	01.03.2020
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП ТПУ	Спицына Любовь Юрьевна	Кандидат экономических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К61	Бальков Николай Ленскийевич		
8К61	Рыбаченко Иван Александрович		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студентам:

Группа	ФИО
8К61	Балыкову Николаю Ленскиевичу
8К61	Рыбаченко Ивану Александровичу

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка платформы психологического тестирования для корпоративных клиентов	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования – веб-приложение, автоматизирующее процессы проведения ручного психологического тестирования для корпоративных клиентов. Область применения – в организациях различного масштаба для формирования благоприятного психологического климата внутри трудовых коллективов
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> – Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032 – 78 – Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03 – Использование персональных данных пользователей регулируется Федеральным законом №152 «О персональных данных»
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	<ul style="list-style-type: none"> – Отклонение показателей микроклимата – Отсутствие или недостаток естественного света – Недостаточная освещенность рабочей зоны – Повышенный уровень электромагнитных излучений
3. Экологическая безопасность:	– Анализ негативного воздействия на окружающую природную среду: утилизация компьютеров, ноутбуков, оргтехники.
4. Безопасность в чрезвычайных ситуациях:	Возможные чрезвычайные ситуации: – Пожар

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП ТПУ	Белоенко Елена Владимировна	Кандидат технических наук		

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8К61	Бальков Николай Ленскийевич		
8К61	Рыбаченко Иван Александрович		

Реферат

Выпускная квалификационная работа содержит 132 страницы, 49 рисунков, 28 таблиц, 18 источников и 2 приложения.

Ключевые слова: психологическое тестирование, информационная система, веб-приложение, архитектура приложения, разработка программного обеспечения.

Цель работы – формирование благоприятного психологического климата в трудовом коллективе и повышение заинтересованности персонала в работе.

В первой главе представлен обзор предметной области корпоративных психологических тестирований.

Вторая глава описывает процесс проектирования платформы психологического тестирования для корпоративных клиентов.

В третьей главе описывается программная реализация платформы психологического тестирования для корпоративных клиентов.

Четвертая глава содержит анализ результатов разработки платформы психологического тестирования для корпоративных клиентов.

Пятая глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», в котором отображены потенциал, планирование и эффективность решения.

Шестая глава представляет собой выполненное задание по разделу «Социальная ответственность», в котором рассмотрены организационно-правовые, производственные и экологические аспекты безопасности, а также безопасность в чрезвычайных ситуациях.

Содержание

Реферат	11
Термины и сокращения.....	16
Введение	18
Глава 1. Обзор предметной области.....	19
1.1 Общая информация	19
1.2 Процесс проведения психологического тестирования.....	20
1.3 Обзор конкурентных решений.....	22
1.3.1 Веб-сайт <i>psyttests.org</i>	22
1.3.2 Система тестирования <i>INDIGO</i>	23
1.3.3 Система тестирования <i>StartExam</i>	24
1.4 Выводы по главе	26
Глава 2. Проектирование платформы психологического тестирования для корпоративных клиентов	27
2.1 Проектирование информационной системы	27
2.1.1 Роли и функциональные возможности пользователей системы	27
2.1.2 Диаграмма BPMN	29
2.1.3 Процесс добавления теста	32
2.2 Проектирование архитектуры веб-приложения.....	34
2.2.1 Отдельные сервера для фронтенда и бэкенда	34
2.2.2 Архитектура клиентского приложения	35
2.2.3 Архитектура RESTful-сервиса	36
2.2.4 Логическая схема базы данных	37
2.3 Выбор инструментов для разработки	39
2.3.1 Бэкенд-сервер	39
2.3.2 Система управления базой данных	41
2.3.3 Фронтенд-сервер	41
2.3.4 Библиотека логирования для фронтенд-сервера	42
2.3.5 Фронтенд-фреймворк	43
2.4 Выводы по главе	43
Глава 3. Программная реализация платформы психологического тестирования для корпоративных клиентов	44
3.1 Фронтенд-сервер.....	44
3.1.1 Структура проекта	44
3.1.2 Каркас приложения.....	46

3.1.3 Система плагинов	47
3.1.4 Управление сессиями	48
3.1.5 Контроллеры	50
3.1.6 Middleware функции	52
3.1.7 Сервисы для доступа к API бэкенд-сервера	52
3.1.8 Логирование	53
3.1.9 Запуск сервера.....	54
3.2 Клиентское приложение.....	54
3.2.1 Структура проекта	54
3.2.2 Компонентный подход разработки	56
3.2.3 Система реактивности	57
3.2.4 Хранилище Vuex	58
3.2.5 Шаблоны страниц	60
3.2.6 Маршрутизация.....	61
3.2.7 Обращение к API фронтенд-сервера.....	62
3.2.8 Модули клиентского приложения	62
3.2.8.1 Лэндинг.....	62
3.2.8.2 Панель администратора	62
3.2.8.3 Панель разработчика.....	63
3.2.8.4 Плагин для прохождения тестирования.....	63
3.2.8.5 Статичные страницы	64
3.2.9 Сборка проекта.....	64
3.3 Бэкенд-сервер.....	66
3.3.1 Описание структуры проекта	66
3.3.2 Общая схема бэкенд-сервера.....	67
3.3.3 Фрагменты кода и его описание.....	68
3.3.4 Генерация excel -отчета	69
3.3.5 Подпроект миграций.....	70
3.4 Выводы по главе	71
Глава 4. Анализ результатов разработки платформы психологического тестирования для корпоративных клиентов	72
4.1 Метрики программного кода	72
4.2 Анализ производительности	73
4.3 Добавление теста «Опросник агрессивности А. Басса и А. Дарки»	74

4.3.1 Методика тестирования	74
4.3.2 Интерпретация результатов теста	75
4.3.3 Добавление теста на платформу тестирования	77
4.4 Демонстрация интерфейса приложения	79
4.4.1 Запуск тестирования	79
4.4.2 Прохождение тестирования	83
4.4.3 Получение отчёта о тестировании	86
4.7 Выводы по главе	86
Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	87
5.1 Введение	87
5.2 Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения ...	88
5.2.1 Предпроектный анализ	88
5.2.2 Потенциальные потребители результата разработки	89
5.2.3 Анализ конкурентных технических решений	90
5.2.4 Оценка перспективности разработки	94
5.2.5 SWOT-анализ	95
5.2.6 Определение возможных альтернатив разработки	98
5.3 Планирование графика работ	98
5.3.1 Участники и структура работ	98
5.3.2 Определение трудоёмкости выполнения работ	99
5.3.3 Разработка графика проведения работ	101
5.4 Планирование и формирование бюджета научных исследований	103
5.4.1 Расчёт материальных затрат научно-технического исследования ..	103
5.4.2 Основная заработная плата исполнителей исследовательской работы	104
5.4.3 Дополнительная заработная плата	106
5.4.4 Отчисления во внебюджетные фонды	107
5.4.5 Накладные расходы	108
5.4.6 Бюджет затрат научно-исследовательского проекта	108
5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	109
5.6 Вывод по главе	112
Глава 6. Социальная ответственность	113

6.1 Введение.....	113
6.2 Правовые и организационные вопросы обеспечения безопасности.....	114
6.2.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.	114
6.2.2 Организационные мероприятия при компоновке рабочей зоны.	115
6.2.3 Особенности законодательного регулирования проектных решений	117
6.3 Производственная безопасность.....	119
6.3.1 Анализ опасных и вредных производственных факторов.....	120
6.3.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов.....	121
6.4 Экологическая безопасность.....	122
6.5 Безопасность в чрезвычайных ситуациях	123
6.5.1 Пожар.....	123
6.6 Вывод по разделу.....	124
Заключение	125
Список литературы	126
Приложение А. Описание классов бэкенд-сервера	129
Приложение Б. Программный код контроллера тестов	132

Термины и сокращения

1. API (Application Programming Interface) – интерфейс, с помощью которого одна компьютерная программа может взаимодействовать с другой программой.
2. Cookie – специфичные для домена файлы, хранящиеся в браузере и передаваемые серверу при каждом исходящем запросе.
3. DOM (Document Object Model) – объектная иерархия HTML-тегов на странице.
4. Express – модуль npm для создания веб-сервера
5. Java – строго типизированный объектно-ориентированный язык программирования.
6. JavaScript – динамически типизированный язык, используемый при веб-разработке.
7. Promise – функция, возвращающая значение асинхронно.
8. PostgreSQL – открытая объектно-реляционная система управления базами данных.
9. TypeScript – статически типизированный язык программирования, выдающий при компиляции JavaScript-код.
10. Бэкенд-сервер – сервер, работающий с базой данных и с которым конечный пользователь общается косвенно, через фронтенд-сервер.
11. Компиляция – процесс преобразования исходного кода из одного формата в другой.
12. Конечный пользователь – человек, использующий готовое приложение через предоставляемый пользовательский интерфейс.

13. Контроллер – класс на сервере, содержащий логику непосредственного взаимодействия с клиентом.
14. Манифест – текстовый файл, декларирующий название и версию пакета, внешние зависимости, скрипты запуска и прочие характеристики прт-пакета.
15. Монтирование – процесс добавления части HTML разметки на страницу.
16. Плагин – связный набор HTML-страниц, JS-кода и CSS-таблиц.
17. Развёртывание – процесс перемещения файлов приложения на вычислительную машину и подготовки их к запуску на выполнение.
18. СУБД (система управления базой данных) – программа, позволяющая выполнять операции для записи, хранения и выборки данных.
19. Фронтенд-сервер – сервер, с которым общается веб-клиент.
20. Хранилище – совокупность программных и или физических элементов для хранения данных.
21. Хук – переопределяемая функция, вызываемая в программе на определённом этапе её выполнения.

Введение

В современном мире коллективный труд приобретает всё большие масштабы. Необходимость умения работы в команде становится стандартным требованием, предъявляемым сотрудникам организаций. При этом одной из основных причин снижения заинтересованности сотрудников организаций в работе является наличие неблагоприятного психологического климата внутри трудового коллектива, членами которого они являются.

Одним из способов налаживания отношений между членами коллектива является проведение корпоративных психологических тестирований. В рамках тестирования выявляются факторы, создающие трудности для взаимодействия членов организации. Зная первопричины проблем, руководству будет проще принять меры по предотвращению и ликвидации факторов формирования неблагоприятного психологического климата.

Однако, реализация традиционного ручного тестирования отнимает много трудового времени. Возникает необходимость в автоматизации процесса проведения психологических тестирований.

Цель разрабатываемой системы – формирование благоприятного психологического климата в трудовом коллективе и повышение заинтересованности персонала в работе.

Для достижения цели поставлены следующие **задачи**:

1. Сделать обзор на тему корпоративных психологических тестирований и программных средств проведения корпоративного психологического тестирования.
2. Спроектировать платформу психологического тестирования для корпоративных клиентов.
3. Разработать платформу психологического тестирования для корпоративных клиентов.
4. Описать результаты проделанной работы.

Глава 1. Обзор предметной области

1.1 Общая информация

Основная цель психологического тестирования – узнать что-либо о личности, которая будет проходить тестирование. Различают несколько методов проведения психологического тестирования [1]:

- Тест-опросник основан на системе заранее отобранных и проверенных с точки зрения их валидности и надежности вопросов, по ответам испытуемых, на которые определенно можно судить об их психологических качествах
- Тест-задание предполагает оценку психологии и поведения человека не на основе того, что он говорит, а на базе того, что он делает. В тестах этого типа человеку дается серия специальных заданий, по итогам, выполнения которых судят об изучаемом качестве
- Проективный-тест проводится посредством проецирования участником действий и ситуаций на другого человека и обычно предназначен для изучения тех психологических и поведенческих особенностей человека, которые им слабо осознаются или вызывают к себе с его стороны крайне отрицательное отношение

Распространенным способом тестирования является комплексное проведение тестов-опросников и тестов-заданий. Взаимодействуя напрямую с психологом, группа людей отвечает на стандартные вопросы, а также выполняют совместные задания. Данные действия позволяют психологу наиболее точно оценить сложившуюся внутри коллектива обстановку. Также довольно часто встречаются проведения только тестов-опросников, по результатам которых специалист может оценить ситуацию и дать рекомендации по улучшению взаимоотношений между членами коллектива.

1.2 Процесс проведения психологического тестирования

Если рассмотреть психологическое тестирование внутри организации в качестве процесса, то началом или инициатором данного процесса является распоряжение руководства о проведении тестирования. Затем сотрудники следуют подготовленным психологом инструкциям. Психолог анализирует полученную в ходе прохождения сотрудниками тестирования информацию и формирует финальный отчёт для руководства. На рисунке 1 представлена диаграмма в нотации IDEF0, которая позволяет понять входы, выходы, механизмы и элементы управления, задействованные в данном процессе.

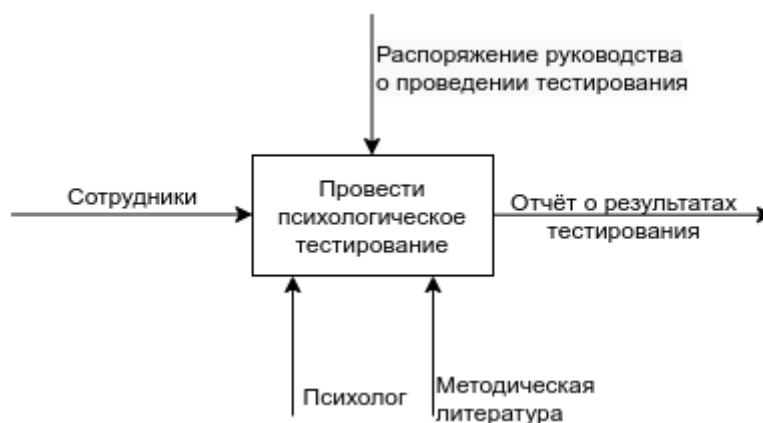


Рисунок 1 – Процесс организации психологического тестирования в виде контекстной диаграммы

Контекстная диаграмма даёт общее понятие о бизнес-процессе. Из контекстной диаграммы можно вынести информацию об общих ограничениях системы, ресурсах ею используемых, о входных и выходных параметрах. Для более детального изучения процесса диаграмма декомпозируется до тех пор, пока процессы на диаграмме не станут атомарными. Декомпозиция контекстной диаграммы представлена на рисунке 2.

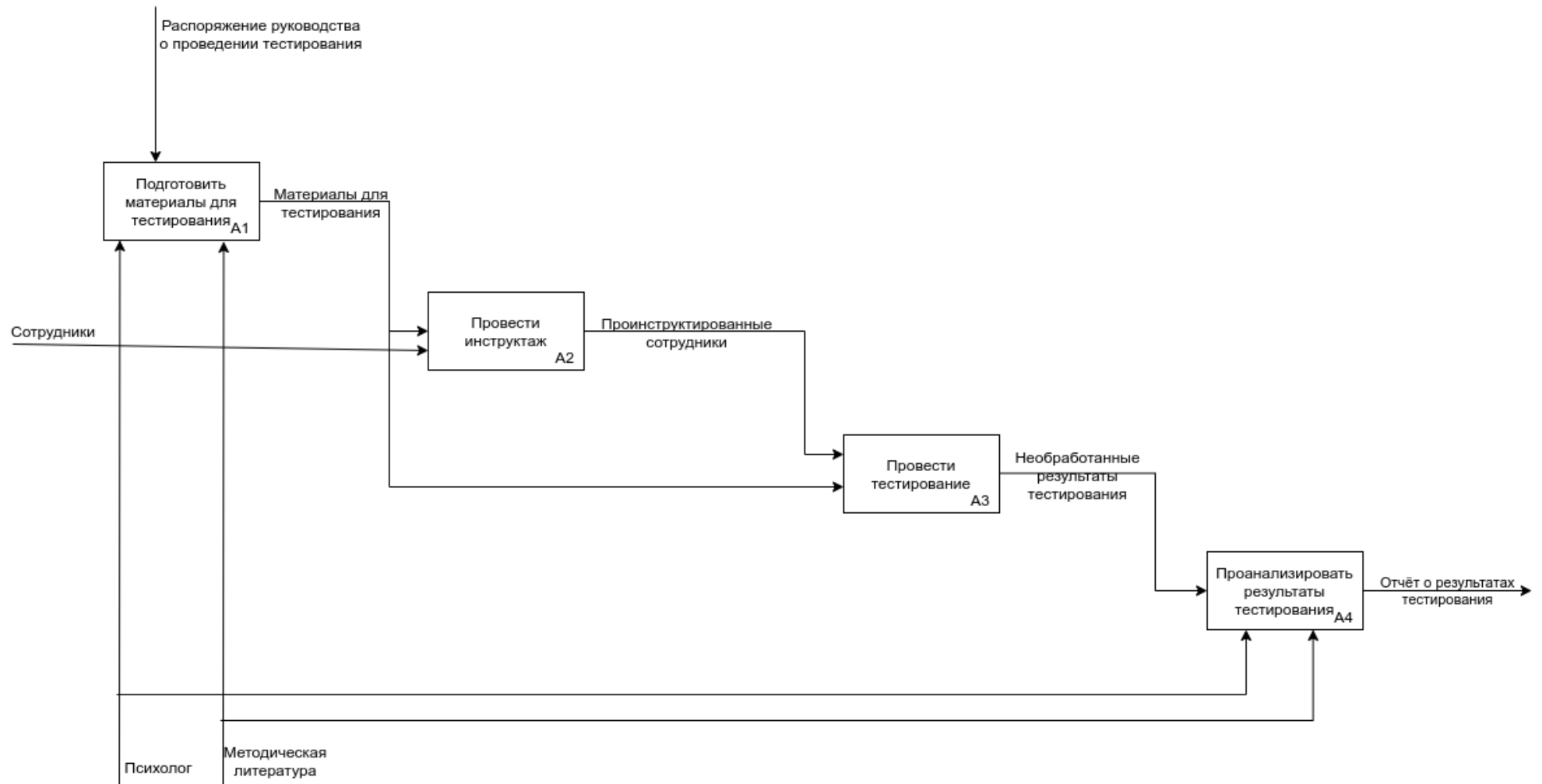


Рисунок 2 – Декомпозиция контекстной диаграммы

Процесс проведения тестирования не является атомарным и представляет интерес для более детального понимания процесса.

Автоматизация процесса проведения психологического тестирования позволит снизить нагрузку на всех членов процесса, включая руководство организации, сотрудников и психолога, что позволит существенно сократить временные и финансовые затраты. Декомпозиция процесса проведения тестирования представлена на рисунке 3.

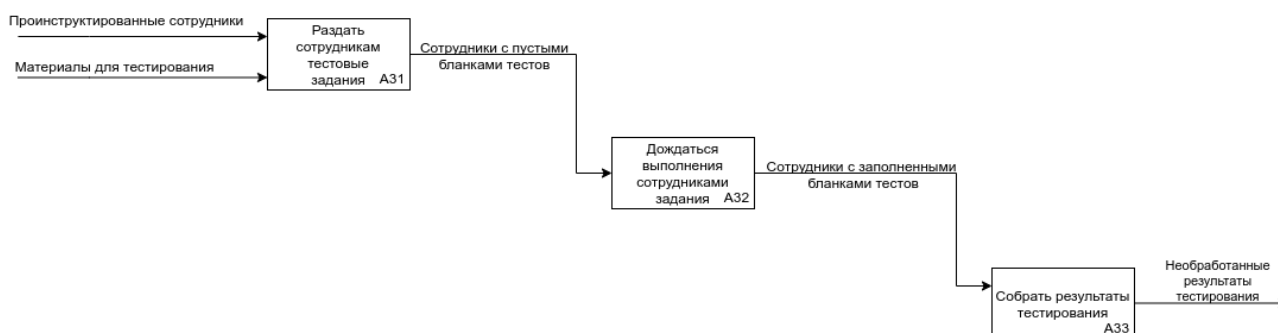


Рисунок 3 – Декомпозиция процесса «Провести тестирование»

1.3 Обзор конкурентных решений

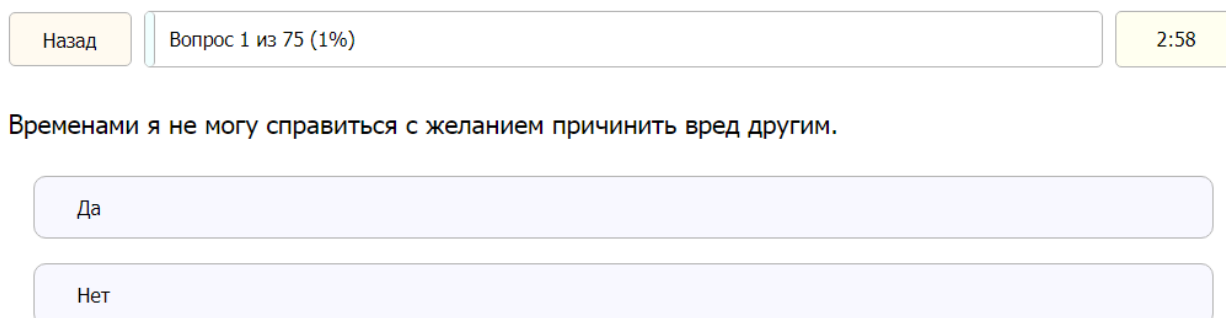
1.3.1 Веб-сайт *psyttests.org*

Веб-сайт *psyttests.org* содержит большую коллекцию психологических тестов [2]. Прохождение тестов является бесплатным и не требует регистрации. После прохождения теста генерируется уникальная ссылка на страницу с анализом результатов тестирования. Доступ к результатам теста может получить любой желающий, имеющий ссылку на этот результат.

Создатели сайта отмечают следующие его основные преимущества: бесплатность, анонимность, удобство интерфейса и высокое качество реализации тестов. К каждому тесту на сайте прилагается подробное описание и список литературы, использованной при его цифровизации.

Сайт аккуратно оформлен в минималистичном стиле. На рисунке 4 представлен скриншот формы выбора ответа на вопрос при прохождении

теста на сайте *psytests.org*. Из рисунка видно, что элементы управления для выбора варианта ответа имеют крупный размер, за счёт чего повышается удобство прохождения тестирования.



The screenshot shows a test interface. At the top left is a 'Назад' (Back) button. In the center, a progress indicator shows 'Вопрос 1 из 75 (1%)'. At the top right is a timer showing '2:58'. Below this, a question is displayed: 'Временами я не могу справиться с желанием причинить вред другим.' (Sometimes I can't cope with the desire to harm others). Below the question are two large, light blue buttons for the answer options: 'Да' (Yes) and 'Нет' (No).

Рисунок 4 – Форма выбора ответа на вопрос при прохождении теста на сайте *psytests.org*

Сайт *psytests.org* не является заменой разрабатываемому решению, поскольку нацелен на проведение персональных тестирований и не содержит средств для организации групповых тестирований.

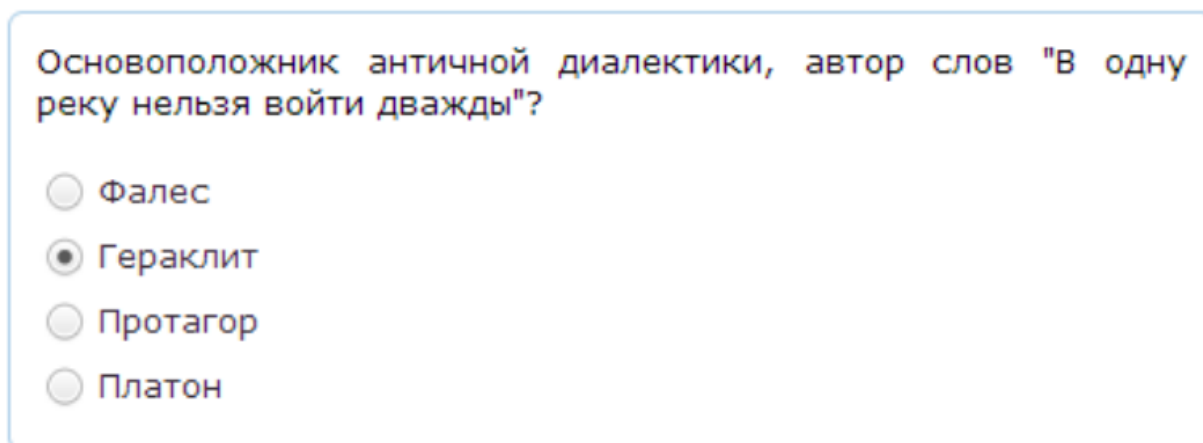
1.3.2 Система тестирования *INDIGO*

Система тестирования *INDIGO* – это профессиональный инструмент автоматизации процесса тестирования и обработки результатов, который предназначен для решения широкого спектра задач [3]:

- 1) Тестирование и контроль знаний учащихся.
- 2) Определение профессионального уровня сотрудников.
- 3) Проведение психологического тестирования.
- 4) Проведение опросов.
- 5) Организация олимпиад и конкурсов.

Система *INDIGO* является мощным и гибким средством для проведения тестирований. Возможна работа с системой как через локальную сеть, так и через Интернет. Имеется встроенный редактор тестов, система правил для прохождения тестов, разработаны средства для повышения безопасности данных.

На рисунке 5 представлена форма выбора ответа на вопрос при прохождении теста на платформе *INDIGO*. Из рисунка видно, что элементы управления оформлены классическим стилем, в следствие чего интерфейс ответа на вопросы у *INDIGO* менее удобен, чем у сайта *psytests.org*.



Основоположник античной диалектики, автор слов "В одну реку нельзя войти дважды"?

Фалес

Гераклит

Протагор

Платон

Рисунок 5 – Форма выбора ответа на вопрос при прохождении теста на платформе *INDIGO*

Основным недостатком системы тестирования *INDIGO* является её высокая стоимость. Стоимость лицензии на данный продукт зависит от количества участников тестирования и может достигать 500000 рублей. Система тестирования *INDIGO* лицензируется по принципу одна лицензия на один сервер тестирования, срок действия лицензии не ограничен и оплата осуществляется только один раз.

1.3.3 Система тестирования *StartExam*

Система тестирования сотрудников *StartExam* является облачным решением [4]. Доступ как к личному кабинету администратора, так и к странице прохождения теста осуществляется через веб-интерфейс. Назначение сервиса – проведение срезов знаний среди сотрудников.

Личный кабинет администратора предоставляет доступ к управлению списком участников тестирования, к редактированию тестов и к управлению

центрами тестирования. Под центрами тестирования понимается группировка участников тестирования, которым нужно пройти один и тот же набор тестов. Участники тестирования также имеют личный кабинет и для прохождения тестирования им требуется вводить свои учётные данные.

На рисунке 6 представлена форма выбора ответа на вопрос при прохождении теста на платформе *StartExam*. Поскольку *StartExam* не специализируется на психологических тестах, картинка взята из теста другой области.

Используемая в интерфейсе палитра цветов является слабо контрастной, поэтому прохождение тестов может оказаться затруднительным для людей с ослабленным зрением.

Какое озеро является самым глубоким озером на планете Земля?

А Танганьика

В Каспийское море

С Байкал

D Иссык-Куль

Рисунок 6 – Форма выбора ответа на вопрос при прохождении теста на платформе *StartExam*

Основным отличием разрабатываемого решения от сервиса *StartExam* является его специализация. Сервис *StartExam* специализируется на тестах для проверки знаний, в то время как разрабатываемое решение на психологических тестированиях.

1.4 Выводы по главе

В данной главе был проведен общий обзор предметной области, рассмотрен процесс проведения психологического тестирования, а также проанализированы существующие программные системы, автоматизирующие процесс проведения психологического тестирования.

Конкурентные решения не могут полностью заменить разрабатываемое в рамках работы решение поскольку часть из них не обладает достаточным функционалом, а часть слишком универсальна и имеет менее удобный интерфейс прохождения теста.

Глава 2. Проектирование платформы психологического тестирования для корпоративных клиентов

2.1 Проектирование информационной системы

2.1.1 Роли и функциональные возможности пользователей системы

На начальном этапе проектирования информационной системы, необходимо определиться с ролями пользователей системы и их функциональными возможностями. Было выявлено три роли пользователей: Администратор, Представитель организации и гость.

На рисунке 7 представлены варианты использования системы Администратором.

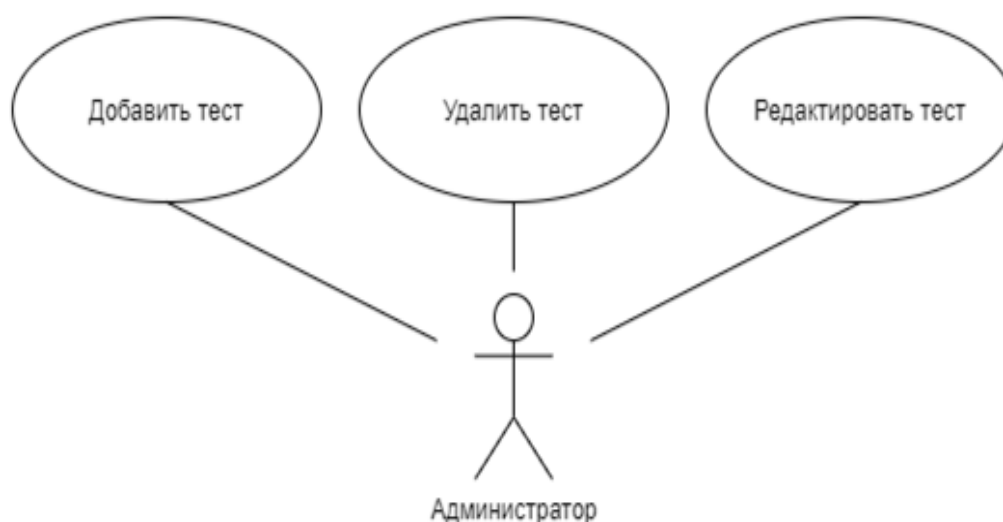


Рисунок 7 – Функциональные возможности администратора

Администратор – авторизованный пользователь, имеющее доступ к панели управления тестами, содержащимися в приложении. Может добавлять, удалять и редактировать тесты внутри системы.

На рисунке 8 представлены варианты использования системы Представителем организации.



Рисунок 8 – Функциональные возможности представителя организации

Представитель организации – авторизованный пользователь, управляющий проведением тестирований со стороны организации. Может просмотреть каталог тестов, запустить тестирование, завершить тестирование и скачать отчёт с результатами проведенного тестирования.

На рисунке 9 представлены варианты использования системы Гостем.

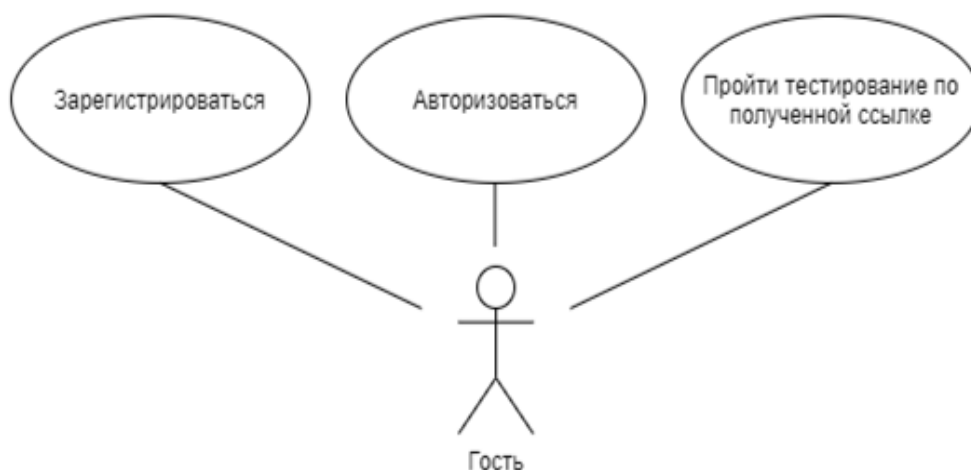


Рисунок 9 – Функциональные возможности гостя

Гость – неавторизованный пользователь. Может зарегистрироваться, авторизоваться и пройти тестирование по ссылке.

2.1.2 Диаграмма BPMN

С помощью диаграммы BPMN описан процесс проведения психологического тестирования после автоматизации данного процесса. Инициатором проведения психологического тестирования выступает руководство компании. В процессе проведения тестирования принимает участие три роли – представитель компании, сотрудник и психолог.

Под представителем компании понимается сотрудник, наделённый дополнительными полномочиями. Представитель компании отвечает за выполнение организационных моментов процесса – он запускает тестирование в системе, следит за тем, чтобы сотрудники прошли тестирование в установленные сроки, контролирует передачу информации между участниками процесса тестирования.

Сотрудником называется участник, чьё психологическое состояние подвергается тестированию. Хотя это никак не обозначено на диаграмме, но на самом деле, может быть несколько сотрудников, участвующих в процессе тестирования. На сотруднике лежит только одна обязанность – пройти тестирование.

Психолог нужен для анализа результатов тестирования. Психолог может являться штатным сотрудником предприятия, а может наниматься по сдельной оплате для анализа результатов отдельных тестирований. Условно процесс проведения тестирования может быть разбит на 5 этапов: запуск тестирования, прохождение тестирования, получение отчёта о тестировании, анализ отчёта о тестировании, передача результатов тестирования руководству.

Диаграмма в нотации BPMN представлена на рисунке 10.

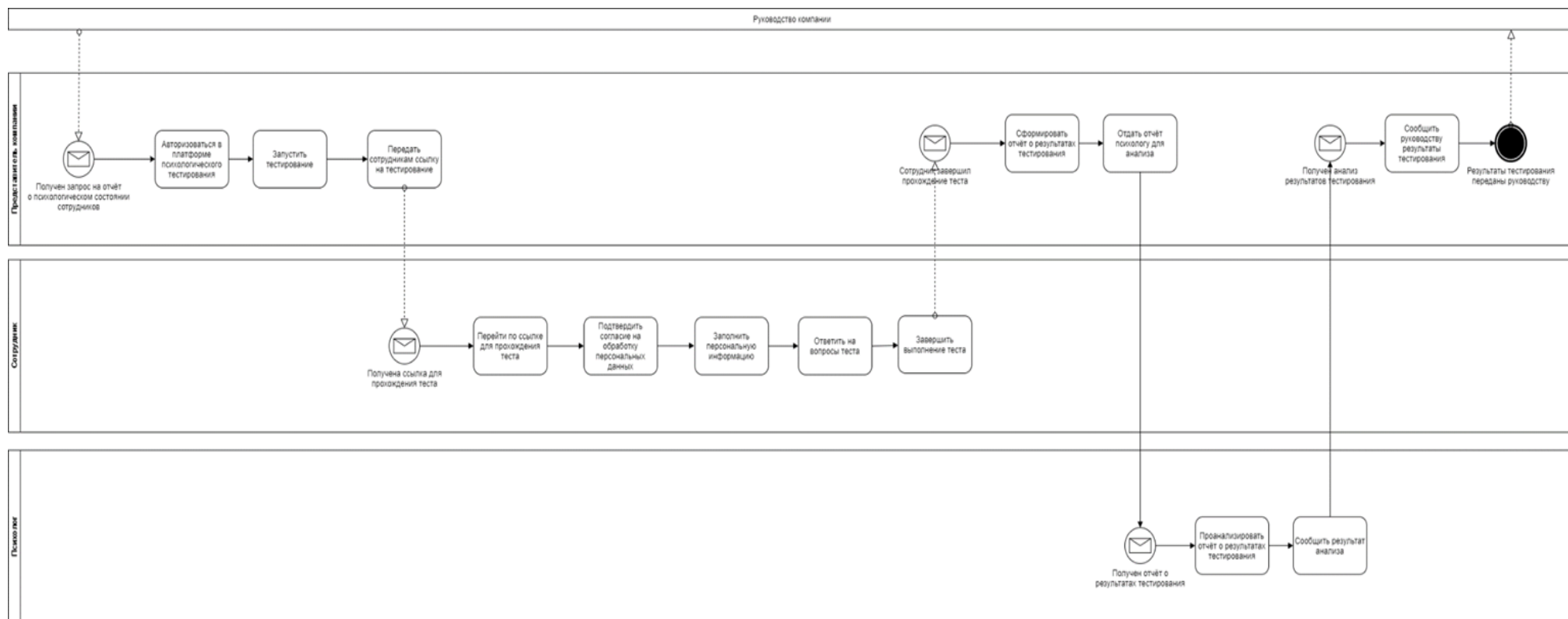


Рисунок 10 – Процесс проведения психологического тестирования после автоматизации

Фрагмент диаграммы BPMN описывающий этап запуска тестирования приведён на рисунке 11.

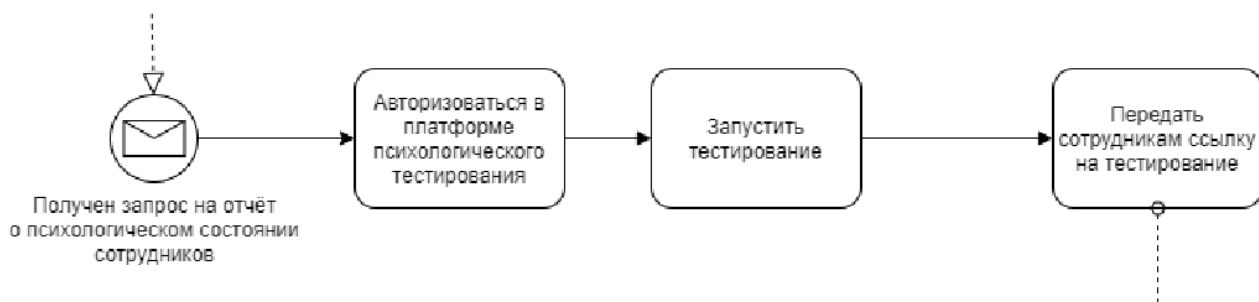


Рисунок 11 – Этап запуска тестирования

На рисунке 12 представлен фрагмент диаграммы BPMN, описывающий прохождение тестирования.

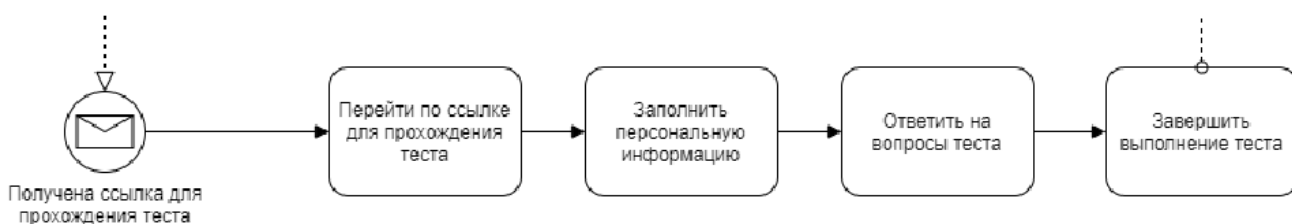


Рисунок 12 – Этап прохождения тестирования

Этап получения отчёта о результатах тестирования представлен на рисунке 13.

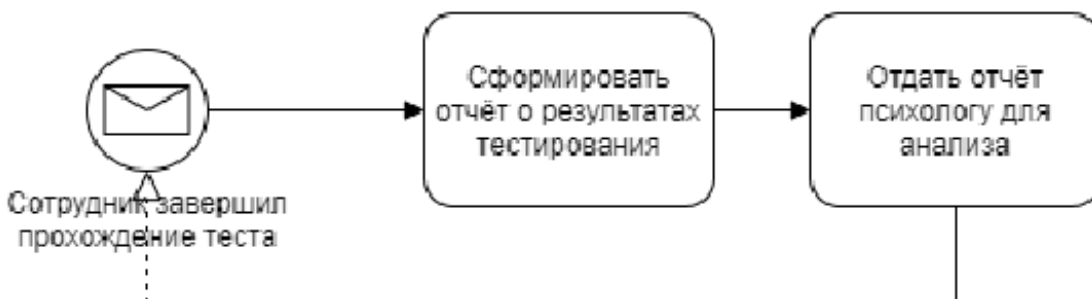


Рисунок 13 – Этап получения отчёта о результатах тестирования

На рисунке 14 представлен этап анализа результатов тестирования.

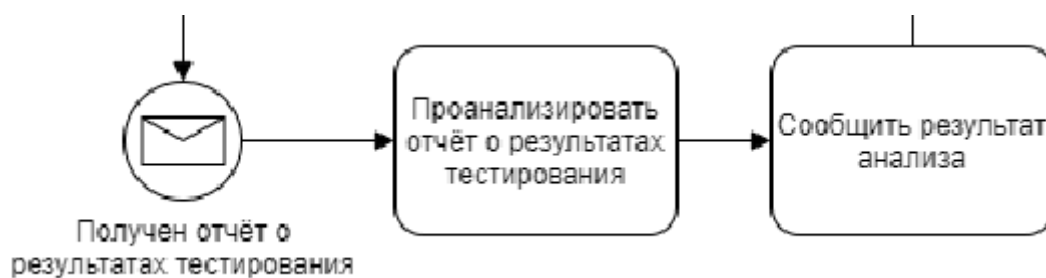


Рисунок 14 – Этап анализа результатов тестирования

На рисунке 15 представлен этап передачи результатов тестирования его инициаторам.

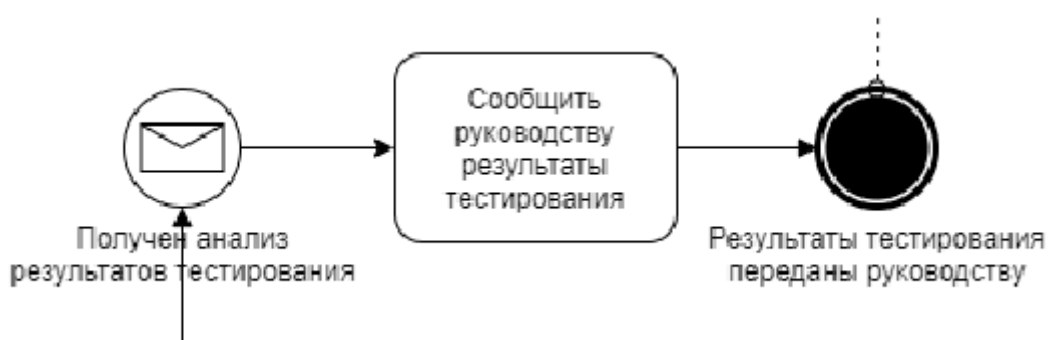


Рисунок 15 – Этап передачи результатов тестирования его инициаторам

Границы разрабатываемой системы охватывают первые три из вышеописанных этапов – запуск тестирования, прохождение тестирования и получения отчёта о результатах тестирования.

Этапы анализа результатов тестирования и передачи результатов анализа руководству в границы системы не попадают. Вместе с этими этапами за рамками системы оказались роли психолога и инициатора тестирования.

2.1.3 Процесс добавления теста

На следующем этапе проектирования был смоделирован процесс внесения администратором нового теста в систему с помощью методологии ЕРС.

Диаграмма процесса внесения нового теста в систему в нотации EPC представлена на рисунке 16. Диаграмма охватывает процесс от состояния «Нужно внести новый тест в систему» до состояния «Тест успешно внесён в систему».

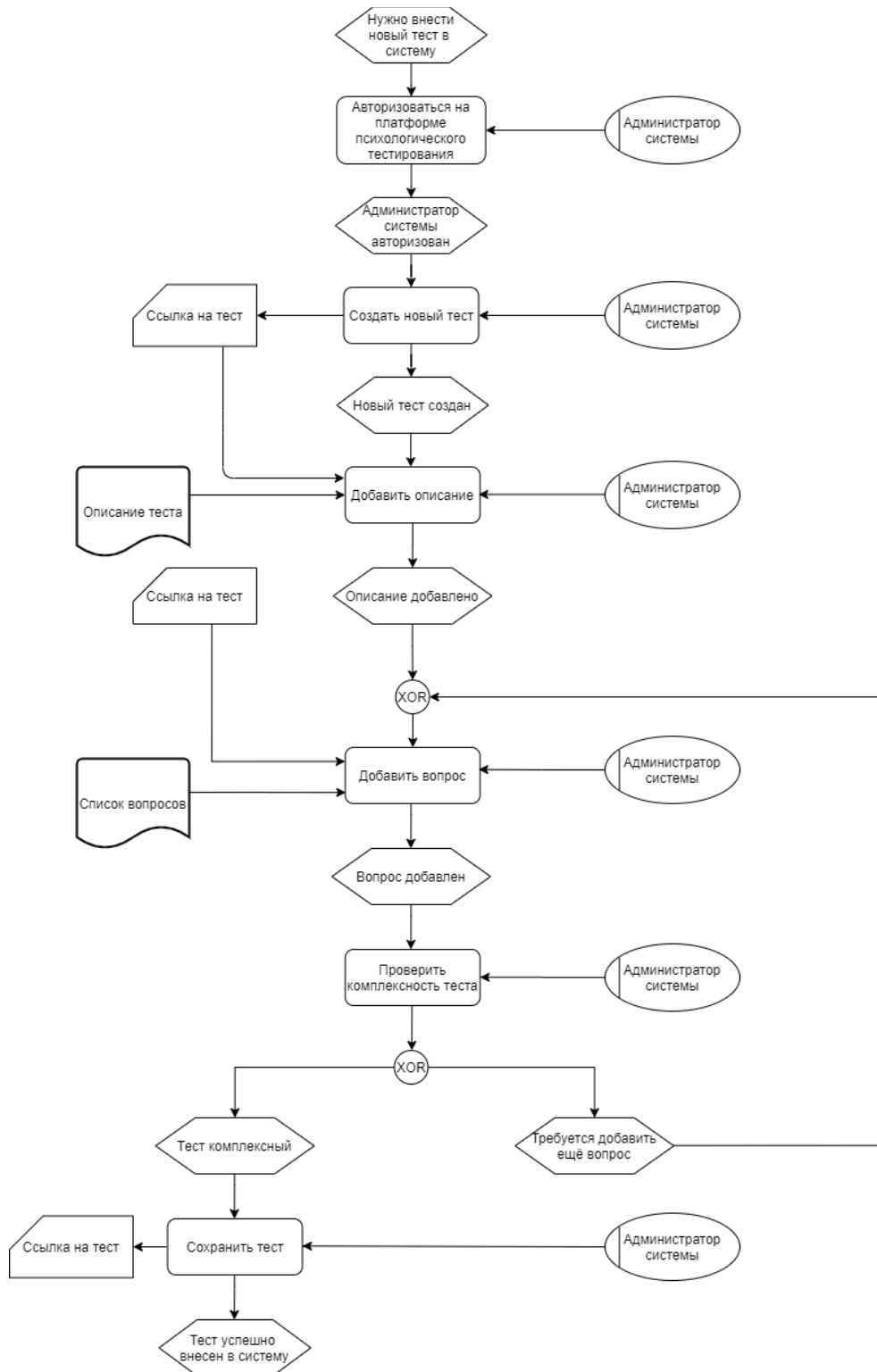


Рисунок 16 – Процесс внесения нового теста в систему

Функция внесения теста добавлена в личный кабинет администратора для того, чтобы не приходилось привлекать дорогостоящих разработчиков для этой задачи. Администратор системы абстрагирован от компании. Функции, возлагаемые на администратора, включают в себя:

- 1) Создание и улучшение тестов
- 2) Обеспечение технической поддержки пользователей

Для выполнения своих обязанностей администратор может контактировать с разработчиками платформы.

2.2 Проектирование архитектуры веб-приложения

2.2.1 Отдельные сервера для фронтенда и бэкенда

Для повышения модульности, безопасности и масштабируемости системы серверная часть приложения разделена на 2 компонента – бэкенд-сервер и фронтенд-сервер. Также разделение приложения на модули способствует упрощению процесса командной разработки. На рисунке 17 изображена схема взаимодействия компонентов веб-приложения.

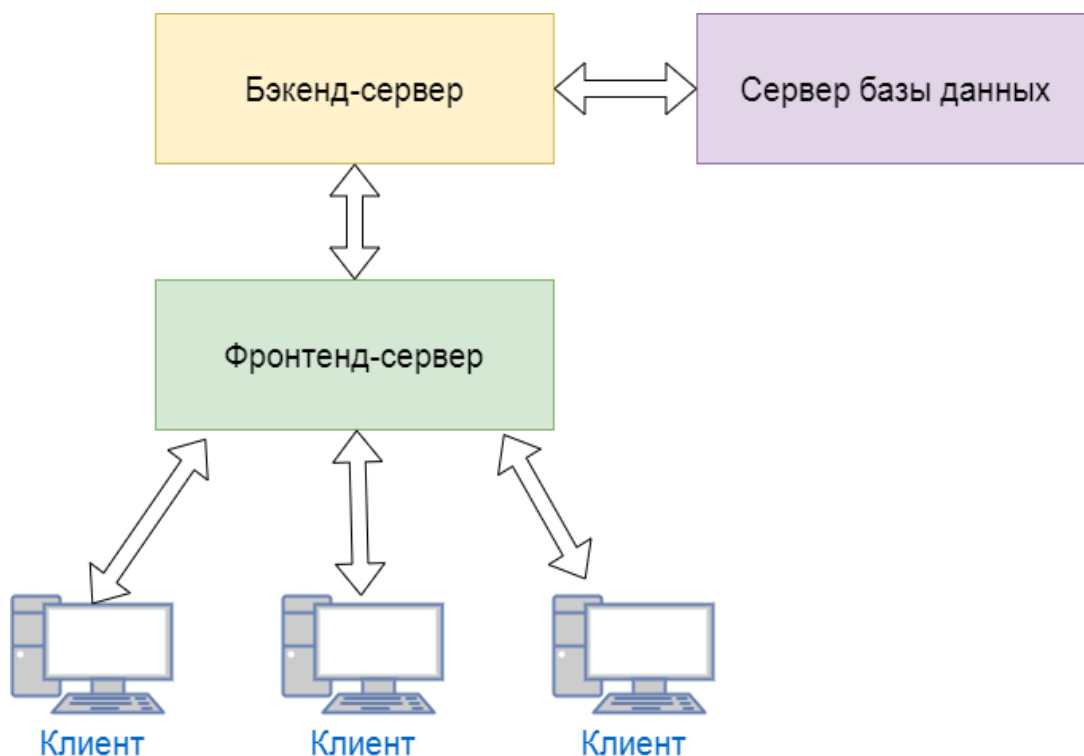


Рисунок 17 – Схема взаимодействия компонентов веб-приложения

Бэкенд-сервер выполняет следующие роли:

- 1) Работа с базой данных
- 2) Выполнение бизнес-логики

Фронтенд-сервер отвечает за следующие задачи:

- 1) Выгрузка пользователю клиентского приложения
- 2) Обработка запросов пользователя
- 3) Управление сессиями пользователей

Стоит отметить, что связь между двумя серверами является однонаправленной. Бэкенд-сервер ничего не знает о существовании фронтенд-сервера, в то время как фронтенд-сервер без доступа к API бэкенд-сервера не может полноценно функционировать.

2.2.2 Архитектура клиентского приложения

Существует три основных подхода для разработки клиентского приложения:

- 1) Одностраничное приложение (SPA)
- 2) Многостраничное приложение (MPA)
- 3) Гибридное приложение (HRA)

Одностраничные приложения характеризуются более отзывчивым интерфейсом, а многостраничные дают меньшую нагрузку на компьютер пользователя. Гибридные приложения комбинируют в себе достоинства и недостатки обеих архитектур.

В данной работе был реализован третий вариант – гибридное приложение. На рисунке 18 показана архитектура гибридного приложения.

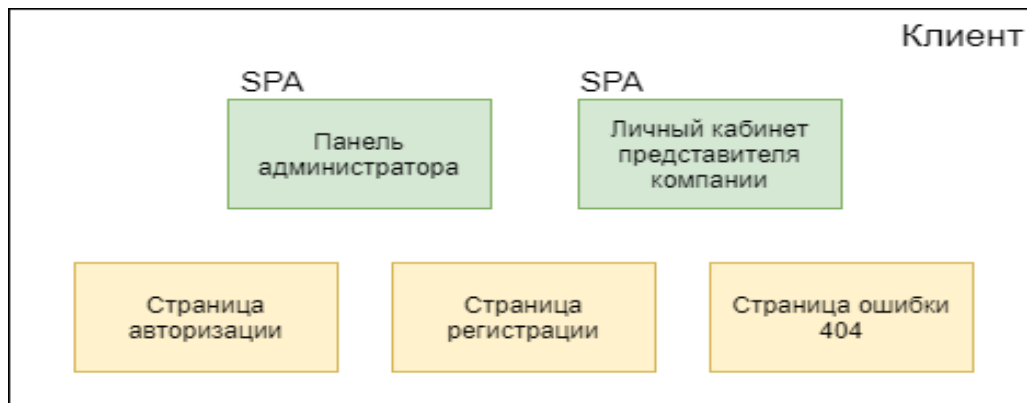


Рисунок 18 – Архитектура гибридного приложения

Панель администратора, лэндинг и личный кабинет представителя компании реализованы как отдельные SPA. Страницы авторизации, регистрации и ошибок также выполнены как отдельные страницы. Наличие нескольких SPA в одном приложении говорит о гибридной архитектуре.

2.2.3 Архитектура RESTful-сервиса

Клиент-серверная архитектура взаимодействия компонентов веб-приложения выполнена в стиле REST. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы и используемых в целях упрощения архитектуры приложения и повышения производительности. Сервер разрабатываемого веб-приложения предоставляет REST API для отправки и получения данных от клиента в формате JSON по протоколу HTTP. Бэкенд-сервер отвечает за получение и обработку HTTP-запросов со стороны клиента. На рисунке 19 изображена архитектура бэкенд-сервера.

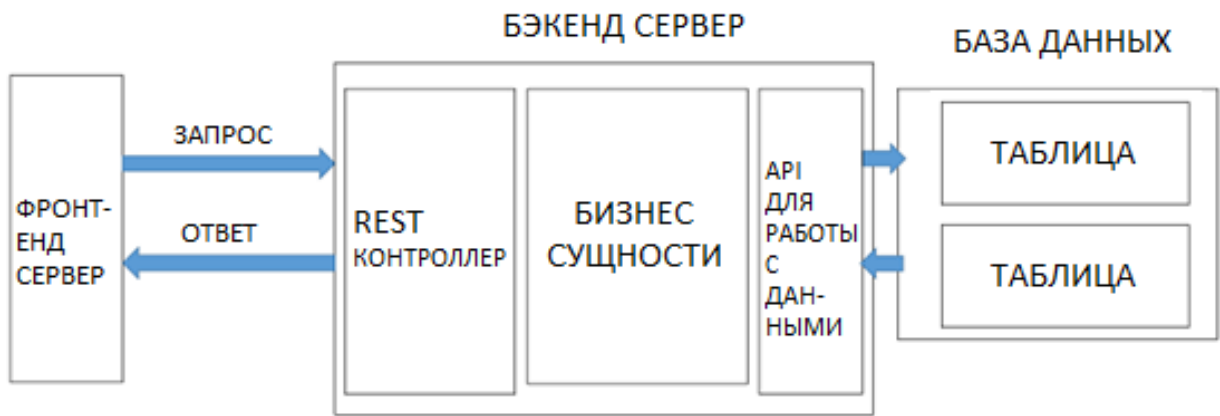


Рисунок 19 – Архитектура бэкенд-сервера

Разработка сервера велась в соответствии с правилами, определяющими архитектуру REST. Основные принципы REST [5]:

- Независимость от состояния – каждый запрос, передающий серверу информацию, содержит все необходимые для выполнения операции данные, а также идентификационную информацию о пользователе.
- Многоуровневая архитектура – архитектура приложения клиент-серверная.
- Единый унифицированный программный интерфейс – соблюдается единый стиль маршрутизации.
- Удобное представление данных – данные представляющие объекты передаются в формате JSON.

2.2.4 Логическая схема базы данных

В ходе проектирования архитектуры веб-приложения была спроектирована логическая схема базы данных. Логическая схема базы данных отличается от физической тем, что отображает модель данных вместе со связями и при этом не зависит от конкретной используемой системы управления базами данных.

Одним из основных средств для проектирования баз данных является методология IDEF1X. Использование широко распространённой нотации позволяет упростить процесс передачи информации об устройстве системы между заинтересованными в этом лицами.

На рисунке 20 представлена логическая схема базы данных, описанная в соответствии с методологией IDEF1X.

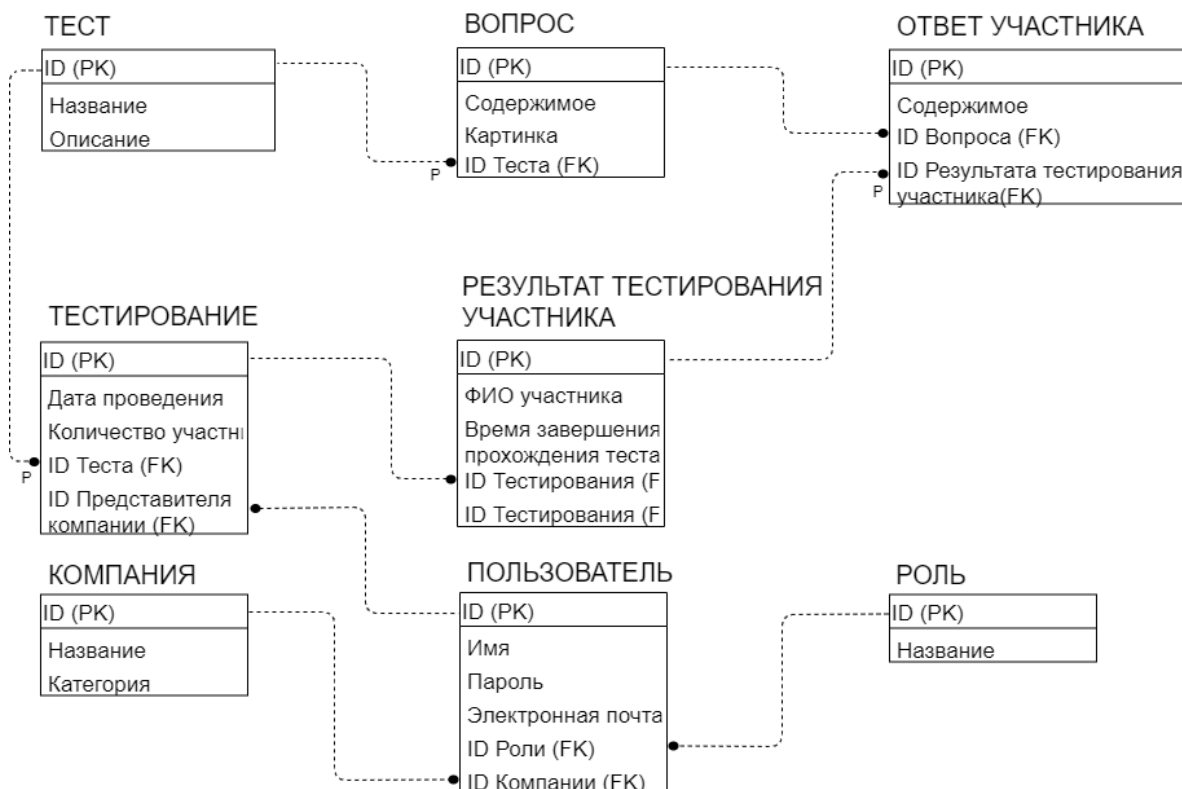


Рисунок 20 – Логическая схема базы данных

В ходе составления диаграммы IDEF1X было выявлено 8 сущностей для хранения в базе данных:

- Сущность «Тест» содержит атрибуты название теста и его описание.
- Сущность «Вопрос» содержит атрибуты содержание, картинка, внешний ключ ID Теста.

- Сущность «Ответ» содержит атрибуты содержимое ответа, внешние ключи: ID вопроса и ID результата тестирования участника.
- Сущность «Тестирование» содержит атрибуты дата проведения, количество участников, внешние ключи: ID теста и ID представителя компании.
- Сущность «Результат тестирования участника» содержит атрибуты ФИО участника, время завершения тестирования и внешний ключ ID тестирования.
- Сущность «Компания» содержит атрибуты название компании и категория.
- Сущность «Пользователь» содержит атрибуты имя пользователя, пароль, электронная почта, внешние ключи: ID роли и ID компании.
- Сущность «Роль» содержит атрибут название роли.

Композиция сущностей «Тест» и «Вопрос» описывает тесты, хранящиеся в системе.

Сущность «Тестирование» является событием, назначаемым представителем организации.

Сущности «Роль», «Пользователь» и «Компания» хранят информацию о пользователях, их роли и компаниях, к которым относятся пользователи.

2.3 Выбор инструментов для разработки

2.3.1 Бэкенд-сервер

Перед началом разработки бэкенд-части приложения было необходимо определиться, какой основной набор технологий использовать. Для этого была составлена матрица морфологического анализа для возможных вариантов, представленная в таблице 1.

Таблица 1 – Определение набора технологий для реализации бэкенд-сервера

Критерий	Весовой коэффициент критерия	Вариант		
		Java + Spring	C# + ASP.Net Core	PHP + Laravel
Опыт работы	0.4	5	4	1
Удобство фреймворков для веб-разработки	0.3	4	5	4
Удобство документации и наличие поддержки сообщества	0.2	5	4	5
Удобство средств разработки	0.1	3	4	3
Итого	1	4.5	4.3	2.9

Основными критериями при выборе набора технологий были: наличие опыта работы с выбранным языком программирования, а также удобство фреймворков для разработки веб-приложений. В качестве языка программирования для серверной части приложения был выбран язык программирования Java. В качестве фреймворка был выбран Spring Framework по причине его популярности и желания разработчика лучше изучить данную технологию.

Spring Boot – представляет собой среду, основная задача которой состоит в упрощении процедур конфигурирования и запуска разрабатываемого приложения.

Spring IOC – ключевой элемент Spring Framework, представляющий собой IOC-контейнер. Осуществляет управление компонентами приложения посредством DI.

Spring Data – представляет собой технологию, обеспечивающую доступ к реляционным и нереляционным данным.

Spring MVC – представляет реализацию шаблона Model-View-Controller в Spring.

Lombok – плагин для компилятора виртуальной машины Java, преобразующий аннотации в Java-код.

Liquibase – библиотека для миграций схем баз данных.

Log4j2 – библиотека для журналирования Java-кода. Удобно используется в связке с Lombok указанием аннотации @Log4j2.

2.3.2 Система управления базой данных

В качестве СУБД используется объектно-реляционная СУБД PostgreSQL. Данная СУБД поддерживает довольно обширное количество типов данных. Кроме числовых типов, типов с плавающей точкой, текстовых типов, булевых типов и других стандартных типов данных, PostgreSQL поддерживает uuid, денежный, перечисляемый, геометрический, бинарный типы, сетевые адреса, битовые строки, текстовый поиск, xml, json, массивы, композитные типы и диапазоны, а также некоторые внутренние типы для идентификации объектов и местоположения логов.

2.3.3 Фронтенд-сервер

В качестве фреймворка для фронтенд-сервера рассматривались *express*, *koa* и *nestjs*. Наиболее старым и популярным из данных фреймворков является *express*. Именно на основе фреймворка *express* построены оба остальных упомянутых фреймворка.

Сравнительный анализ фреймворков представлен в таблице 2.

Таблица 2 – Сравнительный анализ фреймворков для фронтенд-сервера

Критерий	Весовой коэффициент критерия	Вариант		
		express	koa	nestjs
1. Опыт работы	0,4	5	2	1
2. Масштабируемость	0,3	3	3	5
3. Поддержка сообщества	0,3	5	3	4
Итого	1	4,4	2,6	3,1

Фреймворком для разработки фронтенд-сервера был выбран express. В качестве языка программирования для разработки фронтенд-сервера был выбран TypeScript поскольку он обладает рядом преимуществ над JavaScript, основным из которых является возможность статической типизации.

2.3.4 Библиотека логирования для фронтенд-сервера

Логирование играет важную роль при разработке и поддержке системы. Существует большое количество библиотек логирования сильно отличающихся друг от друга функционалом. Морфологическое сравнение библиотек логирования представлено в таблице 3.

Таблица 3 – Сравнительный анализ библиотек логирования для фронтенд-сервера

Критерий	Весовой коэффициент критерия	Вариант		
		log4js	debug	winston
1. Опыт работы	0,2	5	0	0
2. Простота конфигурирования	0,3	4	5	3
3. Наличие различных уровней логирования	0,3	5	2	5
4. Поддержка TypeScript	0,1	5	5	5
5. Производительность	0,1	4	5	4
Итого	1	4,6	3,1	3,3

Для логирования была выбрана библиотека log4js. Основными критериями послужили гибкость и простота конфигурирования.

2.3.5 Фронтенд-фреймворк

В настоящее время существует множество фреймворков для разработки пользовательских интерфейсов. К наиболее популярным относятся React, Vue, Angular. В таблице 4 представлено сравнение фреймворков.

Таблица 4 – Сравнительный анализ фронтенд-фреймворков

Критерий	Весовой коэффициент критерия	Вариант		
		Vue	React	Angular
1. Опыт работы	0,2	5	0	3
2. Популярность использования	0,2	3	5	4
3. Простота разработки	0,3	5	4	4
4. Поддержка TypeScript	0,2	3	3	5
5. Производительность	0,1	5	5	4
Итого	1	4,2	3,3	4

Для разработки пользовательского интерфейса был выбран самый молодой из сравниваемых фреймворк – Vue.

2.4 Выводы по главе

В данной главе были смоделированы и описаны основные бизнес-процессы, связанные с предметной областью. Также была спроектирована архитектура платформы психологического тестирования для корпоративных клиентов, выбраны технологии для программной реализации проекта.

Глава 3. Программная реализация платформы психологического тестирования для корпоративных клиентов

3.1 Фронтенд-сервер

3.1.1 Структура проекта

Фронтенд-сервер отвечает за выдачу клиентского приложения, а также за управление сессиями пользователей. Все запросы, совершаемые конечным пользователем, обрабатываются фронтенд-сервером.

С точки зрения файловой иерархии фронтенд-сервер имеет структуру, представленную на рисунке 21.

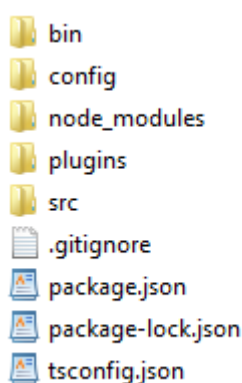


Рисунок 21 – Структура фронтенд-сервера

Папка *bin* содержит скрипты для запуска сервера. В скрипты запуска выносятся информация, зависящая от среды выполнения, такая, например, как порт, занимаемый сервером. Точкой входа в приложение является один из скриптов запуска.

Папка *config* содержит конфигурационные файлы. В частности, к таким файлам относится конфигурация логгера.

Папка *node-modules* является служебной и хранит в себе внешние зависимости проекта. Как правило, эта папка не распространяется вместе с исходным кодом, а восстанавливается из npm-репозитория по зависимостям, указанным в файле *package.json*.

Папка *plugins* служит для установки плагинов. Под плагинами понимаются независимые друг от друга клиентские приложения, которые сервер выдаёт конечным пользователям.

Папка *src* содержит непосредственно исходный код приложения.

Файл *.gitignore* описывает исключения для системы контроля версий GIT.

Файл *package.json* является манифестом проекта.

Файл *package-lock.json* закрепляет какие конкретно версии пакетов использовались при сборке проекта.

Файл *tsconfig.json* служит конфигурацией для TypeScript-компилятора.

Папка *src*, содержащая исходный код сервера, требует отдельного рассмотрения. Она имеет структуру, показанную на рисунке 22.

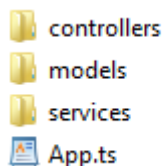


Рисунок 22 – Структура диретокрии *src* сервера

Папка *controllers* содержит контроллеры. Контроллеры – это классы, отвечающие за непосредственное взаимодействие с клиентом. В контроллерах описано API фронтенд-сервера.

Папка *models* содержит модели данных. Модели данных нужны для упрощения обнаружения ошибок при изменении формата передаваемых данных между клиентом и сервером.

Папка *services* содержит сервисы. К сервисам, относятся классы, лежащие на следующем за контроллерами слоем и отвечающие за выполнение бизнес-логики.

Скрипт *App.ts* отвечает за формирование каркаса приложения. Здесь контроллеры, сервисы и *middleware*-функции объединяются в цельное приложение.

3.1.2 Каркас приложения

Ключевым классом сервера является класс *App*. В этом классе все компоненты сервера объединяются в цельное приложение.

Класс *App* по сути своей является обёрткой над библиотекой *express* и в некоторой степени является реализацией паттерна «Декоратор». Необходимость такой обёртки обусловлена требованием изоляции кода. В данном случае весь код, напрямую связанный с настройкой сервера *express*, спрятан внутри одного класса.

Внутри класса *App* содержится 2 приватных поля – *app* и *port*. Переменная *port* хранит номер порта, на котором запускается приложение, а *app* – экземпляр класса сервера *express*, за настройку которого отвечает класс.

Конструктор принимает на вход один объект с параметрами создания сервера. Существует 2 способа передачи параметров в функцию – через один объект-параметр и через несколько параметров. Первый способ является выгоднее с точки зрения качества программирования, поскольку он более гибок и универсален, чем второй. В конструктор передаются следующие параметры:

- 1) номер порта, на котором запускается сервер,
- 2) массив контроллеров, которые нужно установить в сервер
- 3) массив *middleware*-функций, которые должны использоваться сервером

Пример вызова конструктора приведён в листинге 1.

Листинг 1 – Код вызова конструктора

```
const app = new App({
  port: 3000,
  controllers: [
    new PluginController(),
    new AuthController(),
    new UserController(),
    new TestController(),
    new TestingController(),
  ],
  middlewares: [
    bodyParser.json(),
    bodyParser.urlencoded({ extended: true })
  ]
})
```

Благодаря тому, что параметры конструктора передаются в форме одного объекта, в коде создания объекта явно видно, какой из массивов содержит контроллеры, а какой `middleware`-функции.

Внутри конструктора выполняются следующие процессы:

- 1) Создание объекта сервера *express*
- 2) Настройка хранилища сессий пользователей
- 3) Регистрация `middleware`-функций для использования в сервере *express*
- 4) Регистрация контроллеров для использования в сервере *express*
- 5) Подключение плагинов

После того как объект класса *App* был создан, сервер может быть запущен вызовом метода *listen*.

3.1.3 Система плагинов

Сервер выдаёт конечному пользователю страницы. Некоторые из них оформлены как SPA. В рамках данной работы группа логически объединённых страниц называется плагином.

При публикации на сервер плагин упаковывается в обособленную папку. Внутри плагина содержится конфигурационный файл и папка *static*, хранящая

ресурсы, отдаваемые конечному пользователю. К таким ресурсам, в частности относятся HTML-страницы, JS-код, картинки, таблицы CSS. Наличие файла конфигурации является обязательной характеристикой плагина. Файл конфигурации содержит следующие параметры:

- 1) Название плагина. Используется для отображения на странице администратора и при записи логов сервера.
- 2) Роли, необходимые для доступа к плагину. Предотвращают несанкционированный доступ конечных пользователей к плагинам.
- 3) Базовый URL. Это URL, внутри которого будет запускаться плагин. С точки зрения реализации это означает, что данный URL будет добавлен префиксом ко всем маршрутам плагина.
- 4) Маршруты. Сопоставляют URL адрес и относительный путь на HTML страницу, которая должна быть отдана конечному пользователю при посещении данного адреса.

За загрузку плагинов отвечает сервис *PluginLoader*. Ключевым методом данного сервиса является метод *loadPlugins*, который сканирует плагины, имеющиеся в директории *plugins* корневой директории сервера, и регистрирует их в менеджере плагинов. Для сканирования директории *plugins* используются стандартные методы работы с файловой системой из библиотеки “*fs*”. Вызов метода *loadPlugins* осуществляется при старте приложения в классе *App*.

Менеджер плагинов *PluginManager* отвечает за хранение информации об установленных плагинах. В перспективе менеджер плагинов может отвечать за включение и отключение плагинов, за мониторинг их состояния, за изменение их конфигурации, за добавление плагинов на лету и за удаление ненужных плагинов.

3.1.4 Управление сессиями

Сессия используется для идентификации пользователей и хранения, данных, специфичных для каждого отдельного пользователя.

Для реализации механизма сессии используется метод, основанный на работе с файлами куки. Суть этого метода заключается в том, что при обработке запроса от пользователя ему в cookie записывается идентификатор сессии (если он прежде не был записан). Одновременно в хранилище создаётся запись с ключом, равным идентификатору сессии. Таким образом, данные из хранилища сопоставляются с пользователями, характеристиками которых они являются.

Одной из характеристик сессии, основанной на куки, является возможность задать время её жизни. При создании куки указывается срок, через который куки автоматически удаляется. Удаление информации о сессии из базы данных также происходит автоматически.

Для хранения сессий используется in-memory СУБД Redis.

Чтобы сервер express работал с Redis, к нему подключается middleware, содержание которого приведено в листинге 2.

Листинг 2 – Код middleware для подключения express с Redis

```
const redisClient = redis.createClient()
this.app.use(ExpressSession({
  store: new redisStorage({
    host: '127.0.0.1',
    port: 6379,
    client: redisClient
  }),
  secret: 'keyboard cat',
  resave: false
}))
```

При создании хранилища Redis указывается номер порта и сетевой адрес, на котором развёрнута СУБД. Параметр *secret* используется для шифрования файлов куки. Параметр *resave* установлен в *false*, так как это рекомендуемое значение согласно официальной документации.

С точки зрения разработчика middleware-функция *ExpressSession* добавляет атрибут *session* к объекту запроса. Объект *session* используется как для чтения, так и для записи данных сессии. Данные, записанные в объект

session, можно воспринимать как характеристики данные конкретному пользователю.

Со стороны внутренней реализации middleware-функция *ExpressSession* управляет чтением и записью уникального идентификатора сессии в куки, сопоставлением этого идентификатора с объектом запроса пользователя серверу и синхронизацией данных сессии с хранилищем.

Завершение сессии осуществляется путём принудительного удаления файла куки у пользователя.

3.1.5 Контроллеры

Контроллеры предоставляют REST API, которое используется клиентским приложением. Клиентское приложение имеет тот же источник происхождения, что и сервер, таким образом избегается проблема ограничения на кросс-доменный доступ к ресурсам.

Все контроллеры реализуют интерфейс *Controller*, который представлен в листинге 3.

Листинг 3 – Код интерфейса *Controller*

```
export interface Controller {  
  integrateIntoApp(app: Application): void;  
}
```

Интерфейс *Controller* имеет один метод, *integrateIntoApp*, который отвечает за подключение контроллера к серверу. Данный метод принимает на вход один параметр – сервер, в который интегрируется контроллер.

Список контроллеров и задачи, за которые они отвечают, описаны в таблице 5.

Таблица 5 – Контроллеры фронтенд-сервера

Название контроллера	Описание
<i>AuthController</i>	Процессы, связанные с системой авторизации
<i>PluginController</i>	Управление плагинами
<i>TestController</i>	Управление тестами. Просмотр, создание, редактирование, удаление.
<i>TestingController</i>	Управление тестированиями. Просмотр, запуск, остановка, получение отчётов.
<i>UserController</i>	Управление пользователями. Просмотр, создание, редактирование, удаление.

Далее представлен фрагмент контроллера *TestController* в листинге 4.

Листинг 4 – Код контроллера *TestController*

```
export default class implements Controller {
  integrateIntoApp(app: import("express").Application): void {
    app.get('/api/test', (req, res) => {
      TestService.getAllTests().then(response => {
        res.send(response.data);
      }, err => {
        res.sendStatus(404);
      })
    })
    // Описание других API намеренно упущено
  }
}
```

Внутри метода *integrateIntoApp* происходит добавление маршрутов на сервер. В приведённом выше примере создаётся маршрут с адресом */api/test*, обрабатывающий метод GET. Обработчик маршрута делает запрос к бэкенд-серверу и при получении ответа или ошибки, отправляет ответ пользователю.

3.1.6 Middleware функции

Middleware функции выполняются на первом этапе обработки запроса, перед запуском основного обработчика маршрута.

Использовались следующие функции:

- 1) *ExpressSession* – для связывания пользовательских данных из сессии с поступающими на сервера запросами.
- 2) *BodyParser* – для преобразования текстовых данных из тела запроса в формат объекта, удобного для обработки.

3.1.7 Сервисы для доступа к API бэкенд-сервера

Для уменьшения связанности с бэкенд-сервером на фронтенд-сервере создан дополнительный слой классов, вызывающих API бэкенд-сервера. Общение с бэкенд-сервером происходит исключительно через этот слой.

Список сервисов для доступа к API бэкенд-сервера представлен в таблице 6.

Таблица 6 – Сервисы для доступа к API бэкенд-сервера

Название	Описание
<i>TestService</i>	Предоставляет доступ к CRUD тестов
<i>TestingService</i>	Предоставляет доступ к CRUD тестирований

Далее приведён фрагмент кода класса *TestService* в листинге 5.

Листинг 5 – Код класса *TestService*

```
const HOST = process.env.BACKEND_HOST;
export default class {
  static getAllTests(): AxiosPromise<Test[]> {
```

```

        return axios.get(`${HOST}/test`);
    }
    static getTestById(id: string): AxiosPromise<Test> {
        return axios.get(`${HOST}/test/${id}`);
    }
    // Другие функции намеренно пропущены
}

```

Функции, описываемые в данном классе, имеют единственную цель – совершить HTTP-запрос на бэкенд-сервер. Сетевой адрес бэкенд-сервера извлекается из переменной среды и помещается в константную строку. Данные, которые необходимо передать при совершении запроса, поступают в функцию в качестве входных параметров. Функции возвращают *Promise*, что предоставляет при их использовании возможность обработки ошибок.

3.1.8 Логирование

Для логирования используется библиотека *log4js*.

Конфигурация логгера находится в директории *config* и представлено в листинге 6.

Листинг 6 – Конфигурация логгера

```

import { configure } from 'log4js';

export function configureLogger() {
    configure({
        appenders: { console: { type: 'console' } },
        categories: {
            default: { appenders: ['console'], level: 'error' },
            'app': { appenders: ['console'], level: 'debug' },
            'app.pluginloader': { appenders: ['console'], level:
'debug' }
        }
    });
}

```

В конфигурации описаны методы записи и уровни логирования, используемые логгерами.

Доступ к логгеру осуществляется вызовом функции *getLogger* с указанием названия запрашиваемого логгера. Далее приведён листинг 7, демонстрирующий получение логгера.

Листинг 7 – Получение логгера

```
import { getLogger } from 'log4js';  
const log = getLogger('app');
```

Логирование позволяет отслеживать и локализовывать ошибки.

3.1.9 Запуск сервера

Для запуска используется скрипт *start*, описанный в файле *package.json* и приведенный в листинге 8.

Листинг 8 – Содержание скрипта *start*

```
"start": "cross-env PORT=3000 BACKEND_HOST=http://localhost: 8080 ts-  
node bin/start.ts"
```

Используемые переменные среды имеют следующее значение:

- 1) `PORT` – порт, который будет прослушивать сервер
- 2) `BACKEND_HOST` – адрес, по которому доступен бэкенд-сервер

Использование модуля *ts-node* позволяет упростить управление запуском скриптов, написанных на TypeScript. Модуль *ts-node* скрытно компилирует код, написанный на TypeScript, в JavaScript и запускает его.

Выполнение кода начинается со скрипта *bin/start.ts*.

3.2 Клиентское приложение

3.2.1 Структура проекта

Клиентское приложение состоит из нескольких несвязанных между собой модулей. Архитектурное устройство всех модулей примерно одинаковое. В общем виде модуль клиентского приложения имеет структуру директории *src*, как показана на рисунке 23.

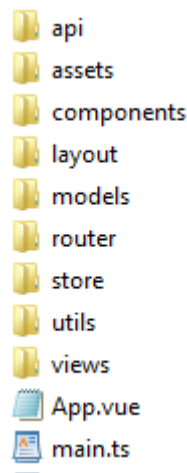


Рисунок 23 – Структура директории src клиента

Папка *api* содержит обёртки над вызовами API-методов бэкенд-сервера. Это делается для того, чтобы абстрагировать реализацию бэкенда от реализации фронтенда.

Папка *assets* содержит статические ресурсы. К таким ресурсам относятся иконки и картинки.

Папка *components* содержит компоненты, которые используются на нескольких страницах.

Папка *layout* содержит общую для нескольких страниц разметку.

Папка *models* содержит описания типов TypeScript для данных, получаемых с бэкенд-сервера или передаваемых на него.

Папка *router* содержит описание маршрутов приложения.

Папка *store* содержит описание Vuex-хранилища.

Папка *utils* содержит функции, не попавшие под другие категории.

Папка *views* содержит сами страницы приложения.

main.ts – это входная точка в приложение с точки зрения исполнения JavaScript кода в браузере.

App.vue – это первичный монтируемый на страницу компонент.

3.2.2 Компонентный подход разработки

Фреймворк Vue подразумевает использование компонентного подхода разработки клиентских приложений. Весь код приложения делится на компоненты. Каждый компонент содержит HTML разметку, JavaScript код и набор CSS стилей. Компоненты имеют специфичный жизненный цикл, в который можно внедрять код для исполнения. Во Vue присутствуют следующие хуки жизненного цикла:

- 1) *beforeCreate* – компонент ещё не был создан
- 2) *created* – компонент создан, инициализирована система реактивности
- 3) *beforeMount* – шаблон компонента скомпилирован и компонент готов к монтированию на страницу
- 4) *mounted* – компонент монтирован на страницу
- 5) *beforeUpdate* – состояние компонента изменено и изменения готовы быть применены к реальному DOM
- 6) *updated* – компонент обновлен

При разработке компонента наиболее часто определяются следующие его свойства:

- 1) *components* – ссылки на компоненты, используемые в данном компоненте
- 2) *data* – внутреннее состояние компонента
- 3) *props* – данные, передаваемые в компонент из родительского компонента
- 4) *computed* – данные, автоматически вычисляемые на основе других данных
- 5) *methods* – функции, используемые компонентом

Компонентный подход уменьшает сложность приложения за счёт изоляции связанных групп кода на HTML, JavaScript и CSS [6].

3.2.3 Система реактивности

Одним из ключевых аспектов фреймворка Vue является система реактивности. К реактивным переменным относятся состояние хранилища Vue, свойства *data*, *props* и *computed* компонента.

При изменении реактивных переменных перерисовывается часть шаблона, зависящая от них, а также пересчитываются *computed* свойства, если на них влияет изменение реактивных переменных.

Принцип работы системы реактивности показан на рисунке 24.

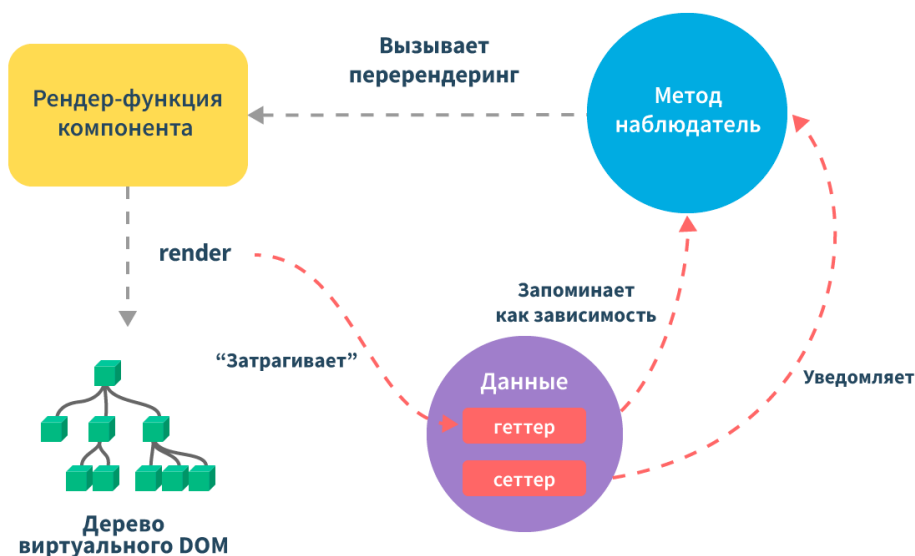


Рисунок 24 – Принцип работы системы реактивности

Использование реактивных переменных значительно упрощает разработку конечного приложения по сравнению с прямым императивным подходом.

3.2.4 Хранилище Vuex

В плагине «Панель администратора» для хранения состояния приложения используется библиотека Vuex.

Хранилище Vuex имеет несколько базовых понятий – состояния, геттеры, мутации и действия.

Состояние хранит данные, контролируемые хранилищем Vuex. В панели администратора состояние описано следующим фрагментом кода, показанном в листинге 9.

Листинг 9 – Состояние хранилища

```
state: {  
  tests: {},  
  testsLoading: false,  
  results: {},  
  testResultsLoading: false,  
},
```

Переменная *tests* представляет из себя объект, ключами которого являются идентификаторы тестов, а значениями сами тесты. Переменная *testsLoading* обозначает происходит ли в данный момент загрузка тестов с бэкенд-сервера.

Аналогичное значение имеют переменные *results* и *testResultsLoading* с тем отличием, что они отвечают не за тесты, а за результаты тестирований.

Геттеры используются для преобразования переменных состояния из одного формата в другой. В данной работе геттеры не использовались.

Мутации используются для изменения состояния хранилища. Все другие способы изменения состояния хранилища, кроме использования мутаций, считаются плохими практиками и не рекомендованы [7]. Мутации, созданные в хранилище модуля «Панель администратора» представлены в таблице 7.

Таблица 7 – Мутации хранилища модуля «Панель администратора»

Название	Описание
ADD_TEST	Добавляет тест в объект, хранящий тесты.
START_TESTS_LOADING	Устанавливает переменную, сигнализирующую о процессе загрузки тестов, в значение true.
STOP_TESTS_LOADING	Устанавливает переменную, сигнализирующую о процессе загрузки тестов, в значение false.
ADD_TEST_RESULT	Добавляет результат теста в объект, хранящий результаты тестов.
START_TEST_RESULTS_LOADING	Устанавливает переменную, сигнализирующую о процессе загрузки результатов тестов, в значение true.
STOP_TEST_RESULTS_LOADING	Устанавливает переменную, сигнализирующую о процессе загрузки результатов тестов, в значение false.

Действия предназначены для высокоуровневого управления хранилищем. В хранилище модуля «Панель администратора» имеется одно действие, отвечающее за инициализацию хранилища. Это действие производит загрузку тестов и результатов тестирований с бэкенд-сервера, параллельно устанавливая переменные, сигнализирующие о статусе загрузки в нужное значение. Вызов действия инициализации хранилища осуществляется при монтировании на страницу самого первого компонента.

3.2.5 Шаблоны страниц

На многих страницах повторяется один и тот же элемент – меню управления. Чтобы не настраивать отображение меню на каждой странице по отдельности, создаётся общий шаблон страниц. Все страницы, реализующие шаблон, содержащий меню, также будут содержать меню.

Для реализации шаблонов используется концепция слотов. Разметка самого шаблона представлена в листинге 10.

Листинг 10 – Разметка шаблона

```
<template>
  <div>
    <Menu />
    <slot />
  </div>
</template>
```

А разметка компонента, использующего шаблон, может выглядеть следующим образом.

Листинг 11 – Разметка использующего шаблон компонента

```
<template>
  <layout>
    <router-view/>
  </layout>
</template>
```

Всё что находится внутри тега *layout* помещается вместо тега *slot* и получается готовая страница. Результат представлена в листинге 12.

Листинг 12 – Результат подстановки

```
<template>
  <div>
    <Menu />
    <router-view/>
  </div>
</template>
```

Таким образом, изменение фона страницы или редактирования ссылок, содержащихся в меню управления, производится только в одном компоненте – шаблоне страниц, а не на каждой странице по отдельности.

3.2.6 Маршрутизация

Маршрутизация реализована с использованием библиотеки Vue-router. В конфигурационном файле маршрутизатора описано, какие маршруты имеются в приложении и какие компоненты нужно отображать по этим маршрутам. Фрагмент конфигурация маршрутов панели администратора представлен в листинге 13.

Листинг 13 – Фрагмент кода конфигурации маршрутов панели администратора

```
const routes: RouteConfig[] = [
  {
    path: '/test',
    redirect: '/test/list',
  },
  {
    path: '/test/list',
    component: () => import(/* webpackChunkName: "about" */
'@/views/test/list/index.vue'),
  },
  {
    path: '/test/:id/start',
    component: () => import(/* webpackChunkName: "about" */
'@/views/test/start/index.vue'),
  },
];
```

На компонент, который монтируется на страницу раньше остальных, добавляется роутер. В результате получается следующий HTML-код, представленный в листинге 14.

Листинг 14 – HTML код после добавления роутера

```
<template>
  <layout>
    <router-view/>
  </layout>
</template>
```

Библиотека Vue-router на основе текущего URL и конфигурации роутера определяет, какой компонент должен отображаться в данный момент времени. Вместо компонента *router-view* подставляется отображаемый на данной странице компонент.

3.2.7 Обращение к API фронтенд-сервера

Для обращения к API фронтенд-сервера создан отдельный слой, аналогичный тому, что сделан на фронтенд-сервере для доступа к API бэкенд-сервера. Далее приведён пример кода в листинге 15, содержащий функции для доступа к API фронтенд-сервера.

Листинг 15 – Пример кода функции для доступа к API сервера

```
const FRONTEND_HOST = 'http://localhost:3000';

export function getAllTestingResults(): AxiosPromise<TestResult[]> {
  return axios.get(`${FRONTEND_HOST}/api/testing/all`);
}

export function startTesting(testId: string): Promise<any> {
  return
  axios.post(`${FRONTEND_HOST}/api/testing/start?testId=${testId}`);
}
```

Использование отдельного слоя для доступа к API сервера, позволяет изолировать этот код от остального приложения.

3.2.8 Модули клиентского приложения

3.2.8.1 Лэндинг

Модуль «Лэндинг» содержит приветственную страницу и реализован в виде одного единственного компонента.

Приветственная страница служит трём основным целям:

- 1) Рассказать о функционале продукта и его отличительных особенностях
- 2) Привлечь новых пользователей
- 3) Перейти в личный кабинет для дальнейшей работы с приложением

Лэндинг-страница должна иметь специальные теги для повышения позиции в поисковой выдаче.

3.2.8.2 Панель администратора

Панель администратора используется администратором и предоставляет следующий функционал:

- 1) Просмотр тестов, доступных для запуска
- 2) Запуск тестирования
- 3) Завершение тестирования
- 4) Просмотр истории проведённых тестирований
- 5) Получение отчёта о результатах тестирования

Для доступа к панели администратора пользователь должен обладать ролью *admin*.

3.2.8.3 Панель разработчика

Панель разработчика предоставляет следующий функционал:

- 1) Создание тестов
- 2) Редактирование тестов
- 3) Удаление тестов
- 4) Просмотр списка работающих на сервере плагинов

Следующий функционал мог быть в панели разработчика, но не был реализован:

- 1) Просмотр списка зарегистрированных пользователей
- 2) Управление плагинами

Для доступа к панели разработчика пользователь должен обладать ролью «root».

3.2.8.4 Плагин для прохождения тестирования

Данный плагин предназначен для пользователей, перешедших по ссылке для прохождения тестирования.

Ссылка на прохождение тестирования имеет формат *http://localhost:3000/test/:id*, где *id* – идентификатор тестирования, для прохождения которого сгенерирована ссылка.

С использованием библиотеки `Vue-router` из URL извлекается идентификатор тестирования и совершается запрос к фронтенд-серверу для получения теста, соответствующего данному тестированию. Полученный в ответ на запрос тест отображается пользователю.

Ответы, выбираемые пользователем, запоминаются в памяти браузера и при завершении тестирования одним запросом отправляются на сервер.

3.2.8.5 Статичные страницы

Статичные страницы небольшие и поэтому были объединены в один модуль. Список статичных страниц выглядит следующим образом:

- 1) Страница ошибки 404
- 2) Страница ошибки 403
- 3) Страница авторизации
- 4) Страница регистрации

Статичные страницы не хранят сложного состояния и поэтому не используют хранилище `Vuex`.

3.2.9 Сборка проекта

Сборка всех модулей клиентского приложения выполняется отдельно. Правила сборки описываются в файле `package.json` в секции `scripts`.

За сборку отвечает скрипт `build`, который имеет вид, представленный в листинге 16.

Листинг 16 – Содержание скрипта `build`

```
"build": "cross-env PLUGIN_DIR=../../server/plugins/admin  
FRONTEND_HOST=http://localhost:3000 BASE_URL=/admin  
NODE_ENV=development npm run _build"  
  "_build": "node prebuild.js && vue-cli-service build"
```

Первая вызываемая в скрипте команда имеет название `cross-env`. Эта команда нужна для того, чтобы сборку можно было производить из-под любой

операционной системы. Дело в том, что способ указания переменных среды в Windows и в Linux различается, а благодаря использованию модуля *cross-env*, возможно универсальное задание переменных среды, независимо от операционной системы.

Для сборки указываются следующие переменные среды:

- 1) `PLUGIN_DIR` – относительный путь к директории, в которую будет помещён результат сборки модуля
- 2) `FRONTEND_HOST` – сетевой адрес фронтенд-сервера
- 3) `BASE_URL` – адрес, по которому плагин публикуется на сервере
- 4) `NODE_ENV` – режим сборки, может принимать значение `production` или `development`

Перед самой сборкой запускается скрипт *prebuild.js*. В модуле «Панель администратора» этот скрипт имеет вид, представленный в листинге 17.

Листинг 17 – Содержание скрипта *prebuild.js*

```
const fs = require('fs');
const glob = require('glob');

if (!fs.existsSync(process.env.PLUGIN_DIR)) {
  fs.mkdirSync(process.env.PLUGIN_DIR);
}

const routes = [];

glob('src/router/**/*.ts', { realpath: true }, (er, files) => {
  files.forEach((fileName) => {
    const file = `${fs.readFileSync(fileName)}`;
    const localroutes = file.matchAll(/path:\s*'(.*)'/g);
    routes.push(...Array.from(localroutes).map((a) => a[1]));
  });
  const _routes = {};

  routes.forEach((route) => {
    _routes[route] = 'index.html';
  });

  const config = {
    roles: ['admin'],
```

```

urlRoot: process.env.BASE_URL,
routes: _routes,
};
fs.writeFileSync(`${process.env.PLUGIN_DIR}/config.json`,
JSON.stringify(config));
});

```

Приведённый выше скрипт сканирует директорию *router* и на основе, содержащихся в ней данных составляет перечень маршрутов приложения. Для поиска файлов в директории *router* используется библиотека *glob*, для извлечения информации из файла – регулярные выражения. Результатом выполнения скрипта является созданный конфигурационный файл в выходной директории плагина.

Для других плагинов перед сборкой запускается аналогичный скрипт с единственным отличием – в перечне ролей, необходимых для доступа к плагину.

3.3 Бэкенд-сервер

3.3.1 Описание структуры проекта

Исходный код бэкенд-сервера сгруппирован в соответствие со структурой, изображённой на рисунке 25.

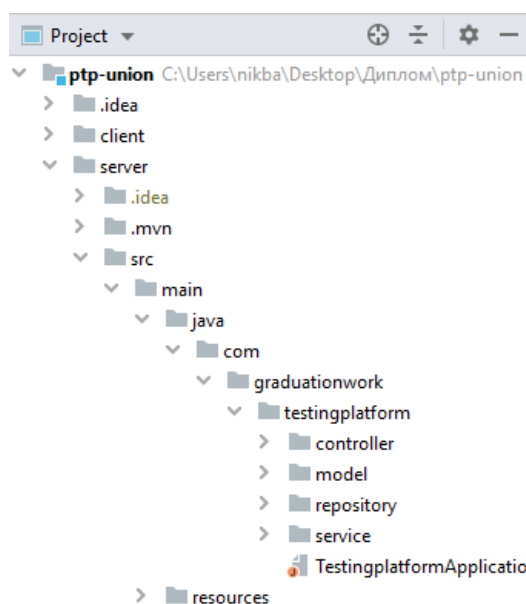


Рисунок 25 – Структура проекта

Папка *controller* содержит классы, принимающие и обрабатывающие запросы, поступающие по определенному маппингу.

Папка *model* содержит классы сущностей приложения.

Папка *repository* содержит классы, отвечающие за хранение, извлечение и поиск данных.

Папка *service* содержит классы фасада для бизнес-логики.

3.3.2 Общая схема бэкенд-сервера

В результате использования выбранных разработчиком бэкенд-сервера технологий, архитектура сервера приобретает вид, показанный на рисунке 26.

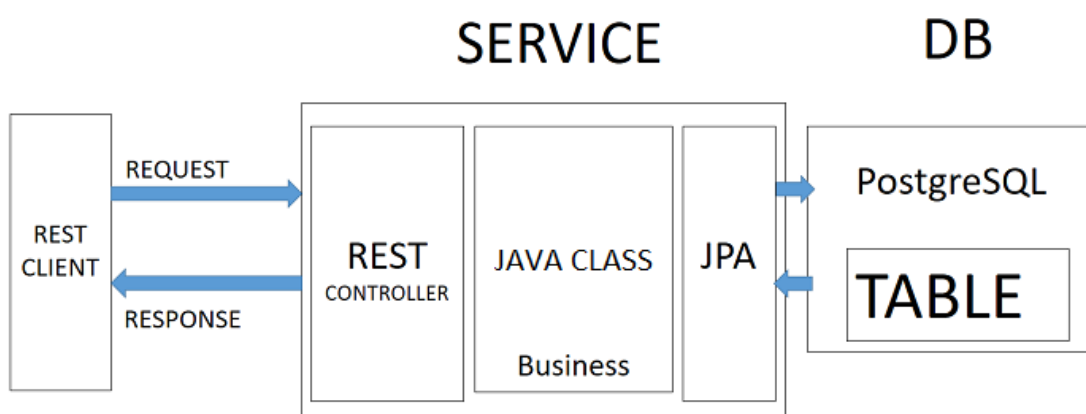


Рисунок 26 – Схема бэкенд-сервера

REST-клиентом выступает фронтенд-сервер, общение которого с RESTful-сервисом происходит посредством HTTP-запросов.

REST-контроллеры сервиса принимают и обрабатывают запросы.

JAVA классы описывают бизнес-сущности веб-приложения.

JPA предоставляет интерфейс взаимодействия с Базой Данных PostgreSQL и осуществляет Object-Relationship-Mapping между объектами и связями Базы Данных и JAVA классами бизнес-сущностей.

3.3.3 Фрагменты кода и его описание

В листинге 18 приведен фрагмент кода класса *TestRepository*.

Листинг 18 – Фрагмент кода класса *TestRepository*.

```
public interface TestRepository extends JpaRepository<Test, Long> {  
}
```

Данный интерфейс-репозиторий наследует интерфейс *JpaRepository*, который предоставляет удобный набор методов взаимодействия с сущностью базы данных, включая необходимые CRUD-операции.

В листинге 19 приведет фрагмент кода класса *TestServiceImpl*.

Листинг 19 – Фрагмент кода класса *TestServiceImpl*

```
@Service  
public class TestServiceImpl implements TestService {  
  
    private final TestRepository testRepository;  
  
    public TestServiceImpl(TestRepository testRepository) {  
        this.testRepository = testRepository;  
    }  
  
    @Override  
    public List<Test> getAllTests() {  
        return testRepository.findAll();  
    }  
}
```

Данный класс содержит в себе зависимость репозиторий-тестов. С помощью паттерна DI в конструкторе внедряется зависимость, а SpringIOC автоматически ей управляет. Также можно наблюдать метод *findAll* репозитория тестов, с помощью которого можно получить все объекты сущности *Test* из базы данных.

В листинге 20 приведен фрагмент кода класса *TestController*.

Листинг 20 – Фрагмент кода класса *TestController*

```
@RestController  
@RequestMapping("test")  
public class TestController {
```

```

private final TestService testService;

public TestController(TestService testService) {
    this.testService = testService;
}

@GetMapping
public List<Test> getAllTests() {
    return testService.getAllTests();
}

```

Данный контроллер реагирует на GET-запрос, поступающий по маппингу `/test`, и выводит список всех тестов. Аннотация `@RestController` указывает, что данные возвращаются в формате JSON.

В листинге 21 приведен фрагмент кода класса `Test`.

Листинг 21 – Фрагмент кода класса `Test`

```

@Entity
@Data
@Log4j2
@Table(name = "test")
public class Test {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private long id;
}

```

Данный класс использует аннотации `@Entity`, `@Table`, указывая, что объекты данного класса содержат данные из полей таблицы `test`. Аннотация `@Data` принадлежит Lombok и дает указания компилятору автоматически сгенерировать Геттеры, Сеттеры, пустой Конструктор и Конструктор со всеми аргументами. Аннотация `@Log4j2` указывает на необходимость журналирования действий и относится к Lombok.

3.3.4 Генерация excel -отчета

Для реализации данной функции была использована библиотека Apache POI. Данная библиотека разрабатывается и распространяется Apache Software Foundation и носит открытый характер. Apache POI включает классы и методы для чтения и записи информации в документы MS Office. Для реализации

отчётов в excel использовался компонент данной библиотеки XSSF (XML Spreadsheet Format).

Ниже представлены классы, используемые для работы с файлами Excel из приложений Java.

- Рабочая книга – HSSFWorkbook, XSSFWorkbook
- Лист книги – HSSFSheet, XSSFSheet
- Строка – HSSFRow, XSSFRow
- Ячейка – HSSFCell, XSSFCell

3.3.5 Подпроект миграций

С помощью библиотеки Liquibase осуществляются миграции базы данных. Данный компонент бэкенд-сервера представлен на рисунке 27 и является обособленным и не входит в общую структуру проекта.

Имя	Дата изменения	Тип	Размер
lib	08.06.2020 19:47	Папка с файлами	
out	08.06.2020 19:47	Папка с файлами	
generateChangeLog	01.06.2020 13:57	Сценарий Windo...	1 KB
initialize	01.06.2020 13:57	Сценарий Windo...	1 KB
liquibase.properties	01.06.2020 13:57	Файл "PROPERTIES"	1 KB
update	01.06.2020 13:57	Сценарий Windo...	1 KB
Инструкция	01.06.2020 13:57	Документ Micros...	4 429 KB

Рисунок 27 – Структура подпроекта миграций

Функционал данного подпроекта представляет собой:

- Генерация файла, содержащего все изменения для репликации схемы Базы Данных. Запускается командным файлом *generateChangeLog.cmd*

- Инициализация схемы Базы Данных. Осуществляется запуском файла *initialize.cmd*
- Добавление новых изменений. Осуществляется запуском файла *update.cmd*

Файл *liquibase.properties* содержит настройки библиотеки для связи с Базой Данных.

3.4 Выводы по главе

В данной главе были описаны особенности программной реализации разрабатываемого приложения. Приложение состоит из 3 компонентов – клиентское приложение, фронтенд-сервер и бэкенд-сервер. Для каждого компонента были описаны файловая структуры проекта и принятые в ходе разработки решения, а также приведены примеры исходного кода с пояснениями.

Глава 4. Анализ результатов разработки платформы психологического тестирования для корпоративных клиентов

4.1 Метрики программного кода

При разработке было использовано несколько языков программирования. Соотношение объёмов использования языков программирования в проекте представлено на рисунке 28. Диаграмма взята из сервиса GitLab.

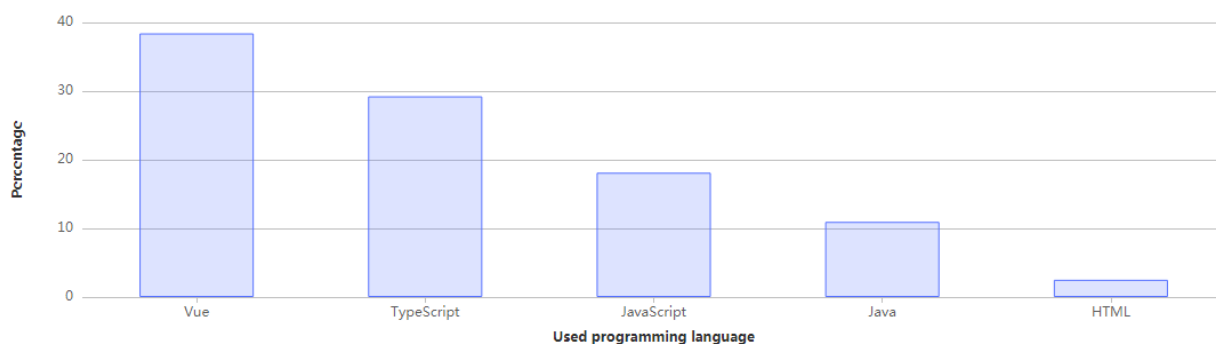


Рисунок 28 – Соотношение использованных языков программирования

В таблице 8 также представлено соотношение языков программирования.

Таблица 8 – Соотношение использованных языков программирования

Язык	Процентное выражение
Vue	38
TypeScript	29
JavaScript	18
Java	11
HTML	2

Высокая доля кодовой базы принадлежит клиентскому приложению. Это связано с тем, что при описании макетов страниц используется очень

короткие строки, содержащие преимущественно по одному HTML тэгу. Строки исходного кода, принадлежащего фронтенд-серверу и бэкенд-серверу более содержательны по своей сути.

Общий объём проекта составил около 110000 строк кода. Для измерения этого показателя использовалась команда *git –numstat*.

4.2 Анализ производительности

При разработке веб-приложений важным показателем является объём передаваемых пользователю данных.

Суммарный размер данных, передаваемых при первичной загрузке страниц, представлен в таблице 9.

Таблица 9 – Размер клиентских модулей при первичной загрузке

Название	Размер, Кбайт
Лэндинг	717
Панель администратора	950
Панель разработчика	949
Плагин для прохождения теста	779
Статичные страницы	757

При вторичной загрузке страниц за счёт кеширования ресурсов браузером, размер передаваемых данных значительно сокращается. В таблице 10 представлены результаты замера объёма передаваемых данных при повторной загрузке страниц.

Таблица 10 – Размер клиентских модулей при повторной загрузке

Название	Размер, Кбайт
Лэндинг	5.6
Панель администратора	5.4
Панель разработчика	5.4
Плагин для прохождения теста	2.8
Статичные страницы	5.6

4.3 Добавление теста «Опросник агрессивности А. Басса и А. Дарки»

4.3.1 Методика тестирования

Опросник враждебности Басса-Дарки был опубликован в 1957 году американскими психологами Арнольдом Бассом и Энн Дарки. Опросник предназначен для выявления уровня агрессивности и враждебности.

Создавая свой опросник, дифференцирующий проявления агрессии и враждебности, А.Басс и А.Дарки выделили следующие 8 видов реакции:

1. Физическая агрессия – это использование физической силы против другого лица.

2. Косвенная агрессия – это агрессия, окольным путем направленная на другое лицо или ни на кого не направленная.

3. Раздражение – это готовность к проявлению негативных чувств при малейшем возбуждении (вспыльчивость, грубость).

4. Негативизм – это оппозиционная манера в поведении от пассивного сопротивления до активной борьбы против установившихся обычаев и законов.

5. Обида – это зависть и ненависть к окружающим за действительные и вымышленные действия.

6. Подозрительность проявляется в диапазоне от недоверия и осторожности по отношению к людям до убеждения в том, что другие люди планируют и приносят вред.

7. Вербальная агрессия – это выражение негативных чувств как через форму (крик, визг), так и через содержание словесных ответов (проклятия, угрозы).

8. Чувство вины выражает возможное убеждение субъекта, что он является плохим человеком, что поступает зло, а также ощущаемые им угрызения совести.

По мнению Басса, реакции может разделить на 2 группы – враждебность и агрессия.

Под враждебностью в данном случае понимается реакция, развивающая негативные чувства и негативные оценки людей и событий. Под агрессией – реакция, в результате которой другой организм получает болевые стимулы. Иными словами, различие между этими понятиями заключается в направленности негативной энергии внутрь самого человека или же на других людей.

В данной работе используется адаптация на русский язык С.Н. Ениколопова, сделанная в 1989 году. Опросник состоит из 75 утверждений, на которые испытуемый отвечает «да» или «нет».

4.3.2 Интерпретация результатов теста

Интерпретация результатов является неотъемлемой частью тестирования. Только после интерпретации результатов можно делать вывод о том находится психологическое состояние человека в норме или нет.

Для интерпретации результата теста ответы на вопросы подвергаются обработке. Опросник агрессивности Басса-Дарки построен таким образом, что каждый вопрос отвечает только за одну исследуемую характеристику. Для каждой из различных форм агрессивных и враждебных реакций подсчитывается количество ответов, данных в пользу поддержки этой формы.

Карта, сопоставляющая вопросы реакциям, за определение которых они отвечают, представлена на рисунке 29. В карте присутствуют номера всех 75 вопросов без повторений.

опро

<i>Реакция</i>	<i>«Да»</i>	<i>«Нет»</i>
<i>Физическая агрессия</i>	1, 25, 33, 48, 55, 62, 68	9, 17, 41
<i>Косвенная агрессия</i>	2, 18, 34, 42, 56, 63	10, 26, 49
<i>Раздражение</i>	3, 19, 27, 43, 50, 57, 64, 72	11, 35, 69
<i>Негативизм</i>	4, 12, 20, 23, 36	—
<i>Обида</i>	5, 13, 21, 29, 37, 51, 58	44
<i>Подозрительность</i>	6, 14, 22, 30, 38, 45, 52, 59	65, 70
<i>Вербальная агрессия</i>	7, 15, 23, 31, 46, 53, 60, 71, 73	39, 66, 74, 75
<i>Угрызения совести, чувство вины</i>	8, 16, 24, 32, 40, 47, 54, 61, 67	—

Рисунок 29 – Интерпретация результатов опросника на уровень агрессивности Басса-Дарки

Физическая агрессия, косвенная агрессия, раздражение и вербальная агрессия вместе образуют суммарный индекс агрессивных реакций, а обида и подозрительность — индекс враждебности.

Враждебность — общая негативная, недоверчивая позиция по отношению к окружающим.

Агрессия — активные внешние реакции агрессии по отношению к конкретным лицам.

Нормой агрессивности является величина её индекса, равная $21 + 4$, а нормой враждебности — $6,5-7 + 3$. При этом обращается внимание на возможность достижения определённой величины, показывающей степень проявления агрессивности.

Агрессивность обладает качественной и количественной характеристикой. Как и всякое свойство, она имеет различную степень выраженности: от почти полного отсутствия до ее предельного развития. Каждая личность должна обладать определенной степенью агрессивности. Отсутствие ее приводит к пассивности, ведомости, конформности и т.д.

Чрезмерное развитие ее начинает определять весь облик личности, которая может стать конфликтной, неспособной на сознательную кооперацию и т.д. Сама по себе агрессивность не делает субъекта сознательно опасным, так как, с одной стороны, существующая связь между агрессивностью и агрессией не является жесткой, а, с другой, сам акт агрессии может не принимать сознательно опасные и не одобряемые формы.

Можно разделить проявления агрессии на два основных типа: первый – мотивационная агрессия, как самооценность, второй – инструментальная, как средство (подразумевая при этом, что и та, и другая могут проявляться как под контролем сознания, так и вне него, и сопряжены с эмоциональными переживаниями (гнев, враждебность).

Практических психологов в большей степени должна интересовать мотивационная агрессия как прямое проявление реализации присущих личности деструктивных тенденций. Определив уровень таких деструктивных тенденций, можно с большой степенью вероятности прогнозировать возможность проявления открытой мотивационной агрессии

4.3.3 Добавление теста на платформу тестирования

Для добавления теста на платформу психологического тестирования используется интерфейс администратора. При создании теста сначала заполняется его название и описание, что продемонстрировано на рисунке 30.

Создание теста

Название	<input type="text" value="Опросник агрессивности А. Басса и А. Дарки"/>
Описание	<input type="text" value="Опросник предназначен для выявления уровня агрессивности и враждебности. Результаты теста позволяют с большой степенью вероятности прогнозировать возможность проявления открытой <u>мотивационной</u> агрессии."/>

Рисунок 30 – Заполнение названия и описания создаваемого теста

После заполнения описательной части администратор переходит к созданию вопросов теста. На рисунке 31 представлена форма наполнения теста вопросами. Количество вариантов ответа на вопрос может быть произвольным, на форме имеются соответствующие кнопки добавления и удаления варианта ответа.

The screenshot shows a form for adding a question to a test. It consists of several elements:

- A question input field containing the text: "1 Временами не могу справиться с желанием навредить кому-либо."
- Below the question, there are two answer options:
 - Option 1: "1 Нет" with a delete button (X) to its right.
 - Option 2: "2 Да" with a delete button (X) to its right.
- A button labeled "Добавить вариант ответа" (Add answer option) is positioned below the options.
- At the bottom left of the form area, there is a button labeled "Добавить вопрос" (Add question).
- At the bottom center, there is a button labeled "Сохранить тест" (Save test).

Рисунок 31 – Форма наполнения теста вопросами

После сохранения теста, он появляется в списке тестов, имеющих на платформе тестирования. На рисунке 32 представлена карточка теста, которую видит администратор в своём личном кабинете.



	<p>Опросник агрессивности А. Басса и А. Дарки</p> <p>Опросник предназначен для выявления уровня агрессивности и враждебности. Результаты теста позволяют с большой степенью вероятности прогнозировать возможность проявления открытой мотивационной агрессии.</p>	
---	--	---

Рисунок 32 – Карточка созданного теста в личном кабинете администратора

Все имеющиеся на платформе тестирования тесты при необходимости могут быть отредактированы администратором.

4.4 Демонстрация интерфейса приложения

4.4.1 Запуск тестирования

Работа с платформой тестирования начинается со стартовой страницы, также называемой лэндингом, представленной на рисунке 33. Стартовая страница содержит описание ключевых особенностей платформы тестирования и ссылки для авторизации и регистрации.

Платформа психологического тестирования

Особенности:

- Предоставляет веб-интерфейс для прохождения теста
- Хранит результаты проведённых тестирований
- Создаёт отчёты о тестированиях для руководящего персонала

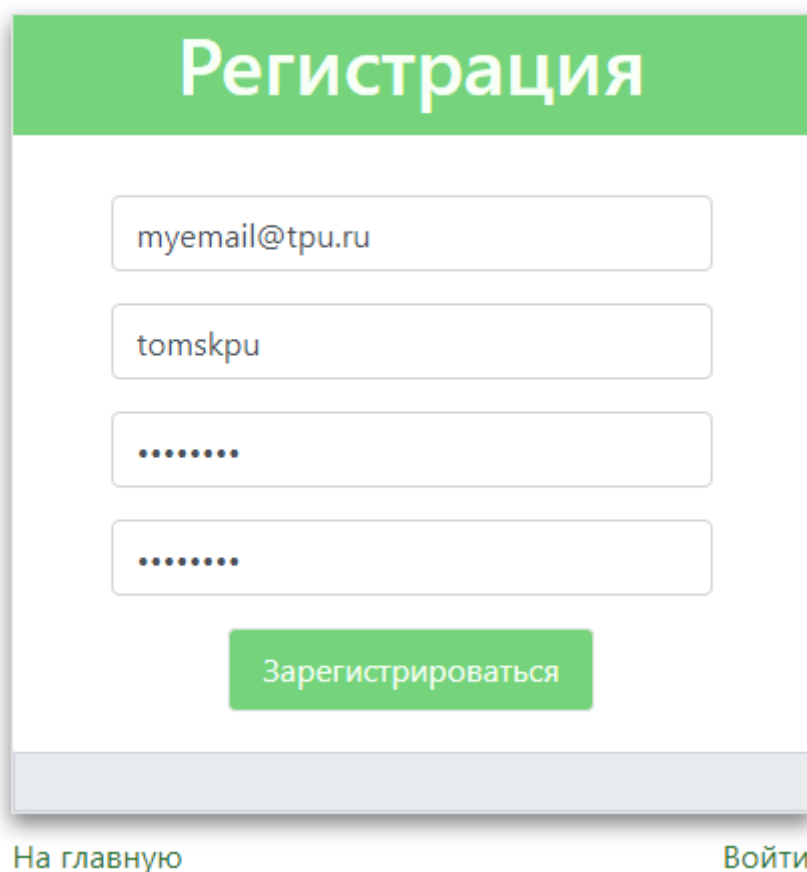
Если вы хотите попробовать сервис в действии, то можете [зарегистрироваться](#).

Если вы уже зарегистрированы, нажмите кнопку ниже, чтобы авторизоваться.

Войти

Рисунок 33 – Стартовая страница платформы психологического тестирования

В случае если пользователь посещает сайт впервые, ему нужно зарегистрироваться. Регистрация осуществляется на отдельной странице и требует указания логина нового аккаунта, адреса электронной почты и пароля. Для устранения фактора опечатки ввод пароля осуществляется 2 раза. Форма регистрации представлена на рисунке 34.



The image shows a registration form with a green header containing the word "Регистрация" in white. Below the header are four input fields: the first contains "myemail@tpu.ru", the second contains "tomskpu", and the next two contain seven dots, representing a password confirmation process. A green button labeled "Зарегистрироваться" is positioned below the input fields. At the bottom of the form, there are two links: "На главную" on the left and "Войти" on the right.

Рисунок 34 – Форма регистрации

При повторной работе с сайтом регистрация не требуется и достаточно авторизации. На рисунке 35 продемонстрирована форма авторизации. В случае неудачной авторизации в нижней строке формы высвечивается соответствующее сообщение.

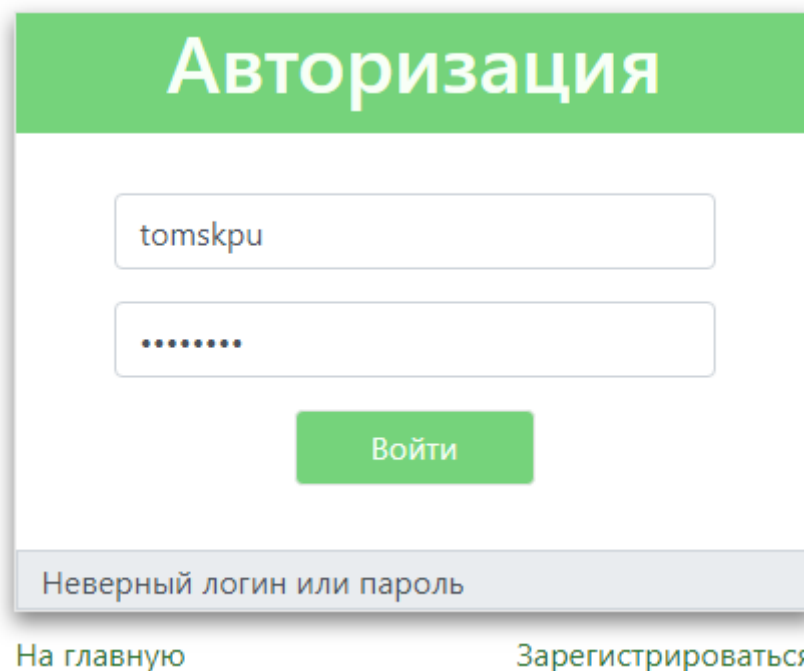


Рисунок 35 – Форма авторизации

В случае успешной авторизации пользователь направляется в личный кабинет. В личном кабинете представитель компании видит список доступных для запуска тестов, что продемонстрировано на рисунке 36.

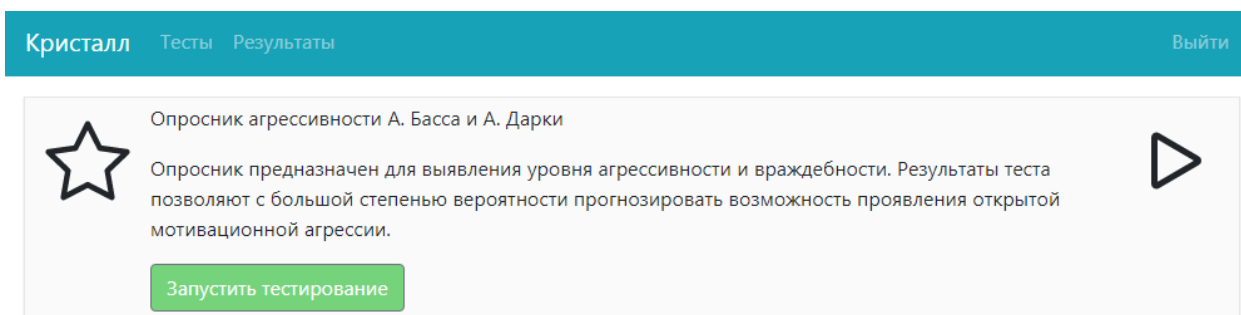


Рисунок 36 – Список тестов, доступных для запуска

Нажатие кнопки «Запустить тестирование» переводит пользователя на страницу запуска тестирования, представленную на рисунке 37. На этом же рисунке видно, что при малом размере экрана верхнее меню навигации изменяет свою форму на мобильную версию.



Опросник агрессивности А. Басса и А. Дарки

Опросник предназначен для выявления уровня агрессивности и враждебности. Результаты теста позволяют с большой степенью вероятности прогнозировать возможность проявления открытой мотивационной агрессии.

После того, как Вы нажмёте кнопку "Запустить тестирование", вам станет доступна ссылка, по которой предполагаемые участники тестирования смогут его пройти. Срок жизни ссылки - одна неделя. Тестирование можно будет завершить досрочно соответствующей кнопкой на вкладке "Результаты тестирования".

Запустить тестирование

Рисунок 37 – Страница запуска теста «Опросник агрессивности А. Басса и А. Дарки»

После запуска теста появляется новая запись на вкладке «Результаты». Выбрав нужный элемент из списка можно получить более подробную информацию о тестировании.

Основными элементами взаимодействия на странице информации о тестировании являются ссылка для прохождения теста, ссылка для загрузки отчёта и кнопка «Завершить тестирование». Ссылку для прохождения тестирования представитель компании передаёт предполагаемым участникам тестирования.

Страница с информацией о запущенном тестировании представлена на рисунке 38.



Опросник агрессивности А. Басса и А. Дарки

Опросник предназначен для выявления уровня агрессивности и враждебности. Результаты теста позволяют с большой степенью вероятности прогнозировать возможность проявления открытой мотивационной агрессии.

Тестирование прошли 2 человека.

Ссылка для прохождения теста: <http://localhost:3000/test/simpledb/50>

[Скачать отчёт](#)

Завершить тестирование

Рисунок 38 – Страница с информацией о запущенном тестировании

После передачи ссылки для прохождения теста его участникам, можно считать тестирование начатым.

4.4.2 Прохождение тестирования

Перейдя по ссылке для прохождения тестирования, его участники попадают на страницу регистрации, на которой содержится краткая инструкция по работе с платформой и форма регистрации на тестирование.

Форма регистрации представлена на рисунке 39. На форме регистрации участник вводит свои фамилию, имя, отчество и подтверждает согласие с условиями обработки персональных данных.

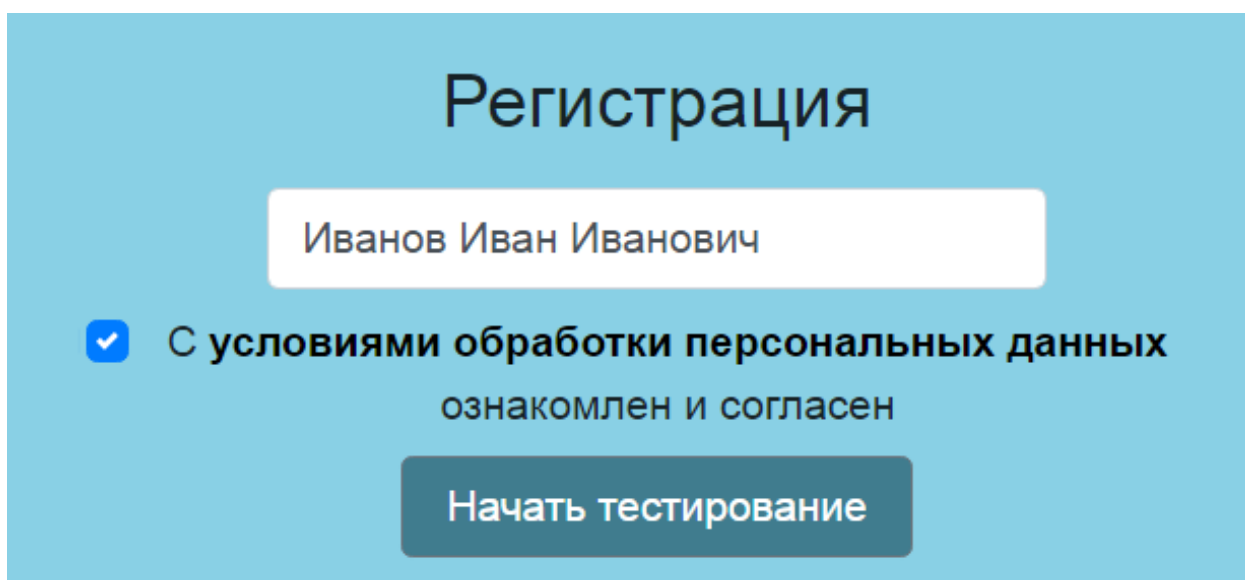


Рисунок 39 – Страница регистрации участника тестирования

Заполнив поле с персональными данными и подтвердив согласие на обработку данных участник тестирования попадает на страницу инструктажа, которая продемонстрирована на рисунке 40.

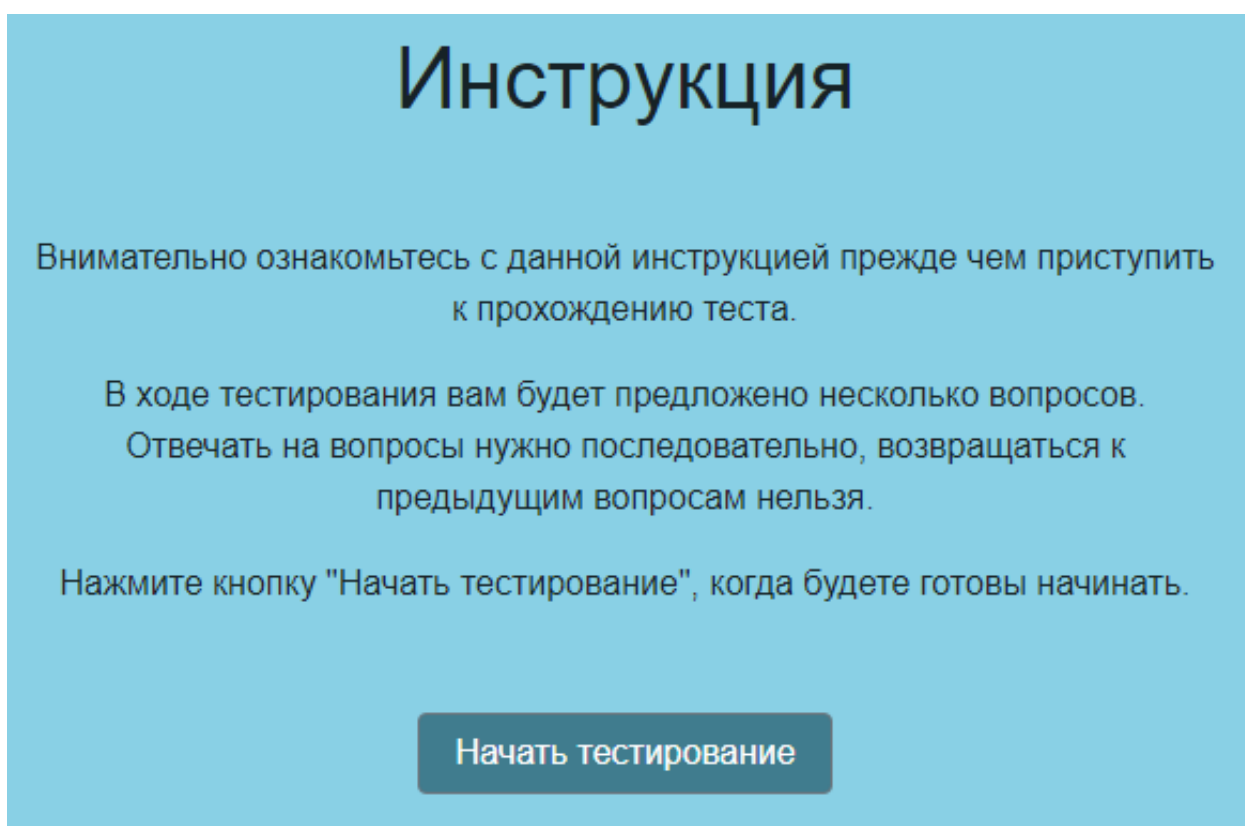
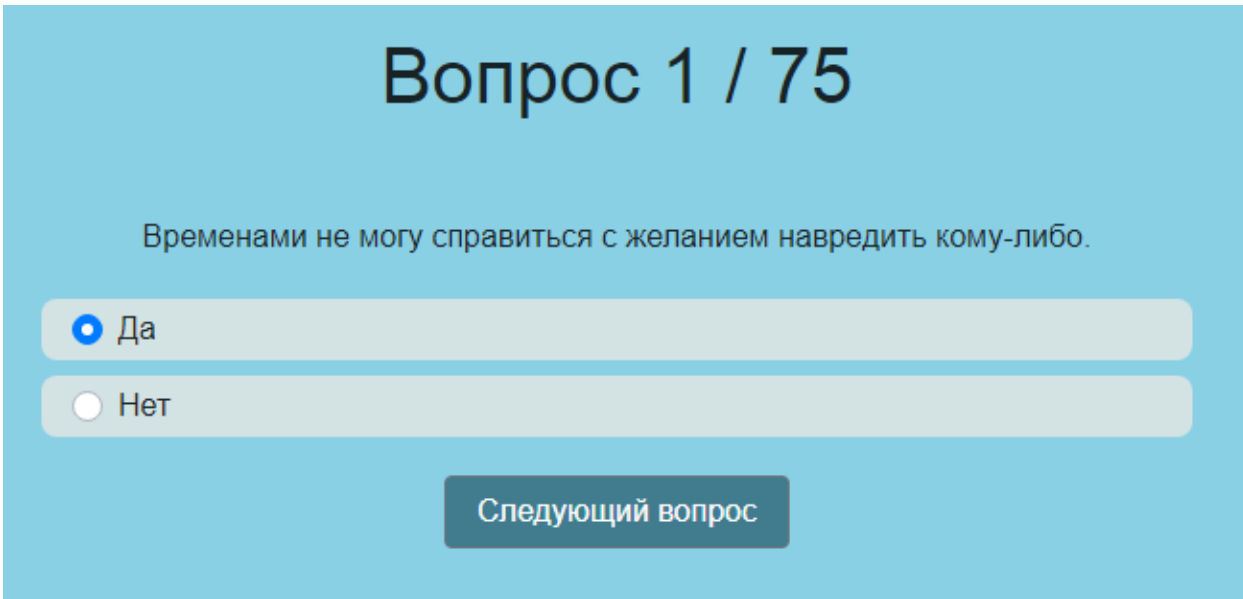


Рисунок 40 – Страница инструктажа к прохождению теста

Ознакомившись с инструкцией к прохождению теста, участник переходит к заполнению ответов на вопросы теста. Заполнение формы ответа на вопрос теста продемонстрировано на рисунке 41.



Вопрос 1 / 75

Временами не могу справиться с желанием навредить кому-либо.

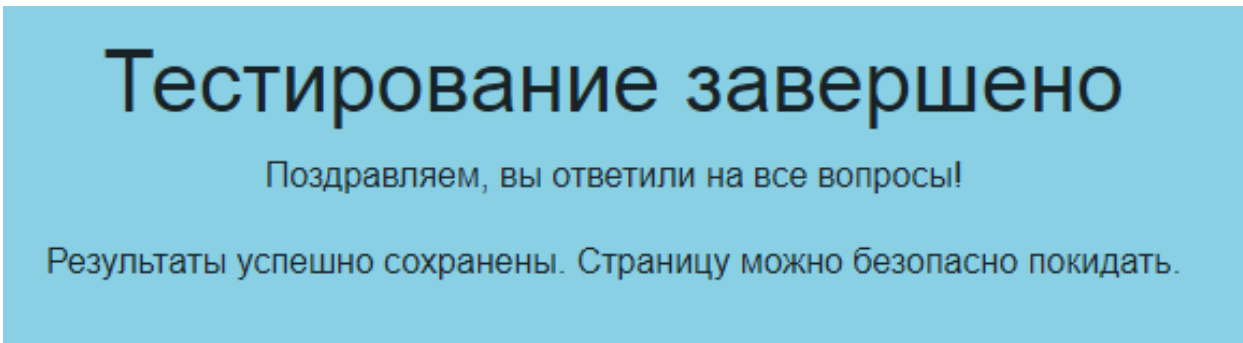
Да

Нет

Следующий вопрос

Рисунок 41 – Заполнение формы ответа на вопрос теста

Окончание прохождения теста информируется соответствующей страницей, представленной на рисунке 42.



Тестирование завершено

Поздравляем, вы ответили на все вопросы!

Результаты успешно сохранены. Страницу можно безопасно покидать.

Рисунок 42 – Страница оповещения о завершении тестирования

После получения сообщения об успешном завершении тестирования, участник тестирования считает свою роль выполненной и может покинуть сайт.

4.4.3 Получение отчёта о тестировании

После того как все участники тестирования завершат его, представитель компании через свой личный кабинет может скачать отчёт о результатах тестирования. Генерируемый отчёт содержит идентификационные данные участников тестирования и полный перечень данных ими ответов. Формат генерируемого отчёта представлен на рисунке 43.

	A	B	C	D	E	F	G
1	ФИО участника	1	2	3	4	5	6
2	Балыков Николай Ленскийевич	2	1	2	2	2	2
3	Рыбаченко Иван Александрович	2	2	2	1	2	2
4							
5							

Рисунок 43 – Отчёт о результатах тестирования

Для интерпретации результатов тестирования отчёт должен быть передан профессиональному психологу.

4.7 Выводы по главе

В главе «Анализ результатов разработки» были даны количественные характеристики полученного в ходе разработки приложения и приведена демонстрация интерфейса приложения. К измеренным характеристикам относятся соотношение использованных языков программирования по проценту использования в коде и объём траффика, получаемого пользователем при загрузке клиентского приложения.

Клиентское приложение получилось тяжеловесным из-за активного использования при разработке сторонних библиотек, однако данный недостаток компенсируется возможностью браузера кешировать загруженные данные.

Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1 Введение

Успешность программного проекта зависит не только от его реализации, но и от эффективности использования ресурсов при разработке. Планирование ресурсов на реализацию проекта является одним из ключевых этапов жизненного цикла программного проекта. Если до старта проекта не заложить достаточное количество ресурсов, то это может повлечь за собой смещение сроков завершения проекта и даже увеличение стоимости разработки. Как следствие при неправильном планировании ресурсов заказчик остаётся не удовлетворён работой, а репутация разработчика оказывается повреждённой.

Не менее важным фактором, влияющим на успешность проекта, является его актуальность и конкурентоспособность. Разработка невостребованного продукта в виду сложности поиска для него покупателя не может являться экономически эффективной. Также, для того чтобы продукт был интересен покупателю, он должен выгодно выделяться своим функционалом на фоне схожих решений.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережения» является проектирование и создание конкурентоспособной программной системы, отвечающей современным требованиям в области ресурсоэффективности и ресурсосбережения.

Для достижения поставленной цели был сформирован следующий список задач:

- 1) оценка коммерческого потенциала и перспективности разработки;
- 2) определение возможных альтернатив проведения разработки, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения;
- 3) планирование сроков, очередности и продолжительности работ, задействованных в разработке;

4) определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности разработки.

Для решения данных задач была дана оценка конкурентоспособности решения, проведён SWOT анализ работ, составлен и визуализирован в виде диаграммы Ганта график работ, проведён расчёт экономической эффективности разрабатываемого решения, вычислены минимальные затраты на реализацию проекта.

5.2 Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения

5.2.1 Предпроектный анализ

Иметь благоприятный климат внутри компании очень важно для повышения эффективности труда сотрудников. Психологический климат зависит от многих показателей, среди которых можно выделить такие как однородность группы, отношения с начальством, личные переживания сотрудников. Оценка уровня компетенций и психологического состояния сотрудников в некоторой степени позволяет отслеживать их удовлетворённость рабочей обстановкой.

Одним из самых быстрых и простых способов оценки психологического состояния коллектива является проведение тестового опроса. Такая форма оценки является наименее отвлекающей для сотрудников, однако если проводить тестирования вручную, на анализ и систематизацию результатов потребуется большое количество сил.

Предметом данного исследования является процесс организации тестирований.

Целью работы является формирование благоприятного психологического климата в трудовом коллективе и повышение заинтересованности персонала в работе.

Для достижения цели сформирован следующий список задач:

- 1) Сделать обзор на тему корпоративных психологических тестирований.
- 2) Сделать обзор существующих на данный момент программных средств автоматизации процесса проведения корпоративного психологического тестирования.
- 3) Спроектировать платформу психологического тестирования для корпоративных клиентов.
- 4) Разработать платформу психологического тестирования для корпоративных клиентов.
- 5) Описать результаты проделанной работы.
- 6) Сделать вывод по проделанной работе.

5.2.2 Потенциальные потребители результата разработки

Проведение тестирований, направленных на исследование группы людей, может потребоваться как физическим лицам, так и организациям.

Цели проведения тестирования могут быть весьма разнообразны. Образовательные организации проводят тестирования обучающихся в целях наблюдения за их успеваемостью и аттестации результатов обучения. Коммерческие предприятия проводят тестирования для определения квалификации сотрудников, а также для получения отзывов от потребителей. Для физического лица наиболее характерным случаем проведения тестирования является опрос его знакомых с целью принятия решения по сложному вопросу.

Результат сегментирования рынка услуг по проведению групповых тестирований представлен в таблице 11. Количеством плюсов обозначена востребованность конкретного вида тестирования заказчиком. Для каждого вида тестирования указан имеющийся уровень конкуренции на рынке.

Таблица 11 – Карта сегментирования рынка услуг по проведению групповых тестирований

		Вид тестирования		
		Психологическое тестирование	Тестирование знаний и компетенций	Опросное тестирование
Заказчик тестирования	Коммерческая организация	+	++	+
	Образовательная организация	+	+++	+
	Физическое лицо			+
Уровень конкуренции		*	***	***

Наиболее привлекательным для внедрения новой системы проведения тестирований является сегмент психологических тестирований, поскольку конкуренция в данном сегменте наименьшая по сравнению с остальными.

Для развития продукта в будущем наиболее привлекательным сегментом является рынок тестирования знаний и компетенций, поскольку соответствующий этому сегменту столбец содержит максимальное суммарное количество плюсов.

5.2.3 Анализ конкурентных технических решений

Наиболее популярными альтернативными решениями для организации тестирований являются система тестирования «INDIGO» и система тестирования сотрудников «StartExam».

Система тестирования «INDIGO» позиционируется как универсальный инструмент для проведения тестирований любого вида. Система хорошо масштабируема, возможно развёртывание внутри предприятия, что исключает риск утечки персональных данных.

Интерфейс программы, изображенный на рисунке 44, визуально сложен в виду наличия широкого функционала, но имеется активная техническая поддержка, компенсирующая данный недостаток.

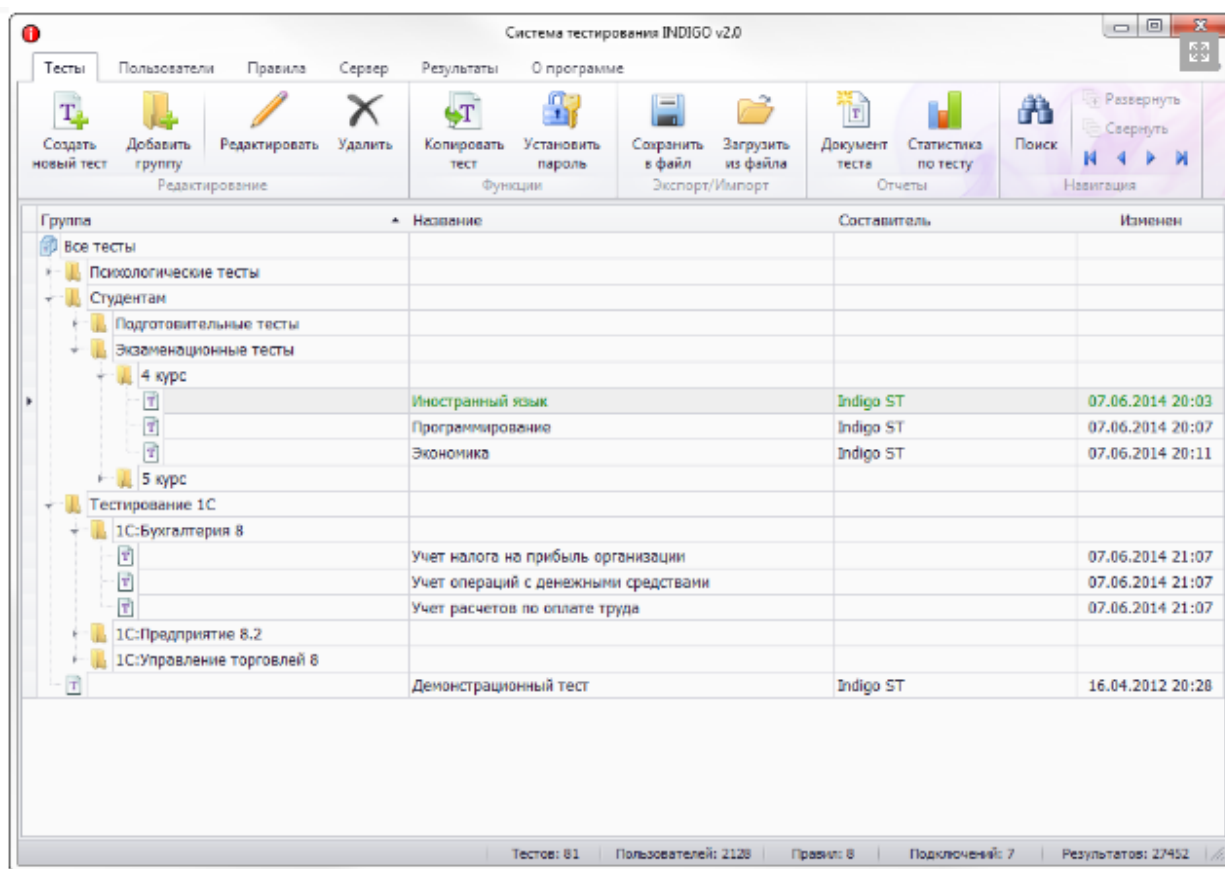


Рисунок 44 – Платформа тестирования INDIGO

Система тестирования «INDIGO» разработана в России и одобрена Минкомсвязью России, что делает её привлекательной для внедрения в государственные структуры.

Система тестирования сотрудников «StartExam» также является отечественной разработкой. Это облачное решение ориентировано на проведение срезов знания. Интерфейс выполнен в минималистичном стиле и представлена на рисунке 45. «StartExam» используется в ряде крупных компаний, в число которых входят крупнейшие игроки российского интернет-рынка Яндекс и Mail.ru Group.

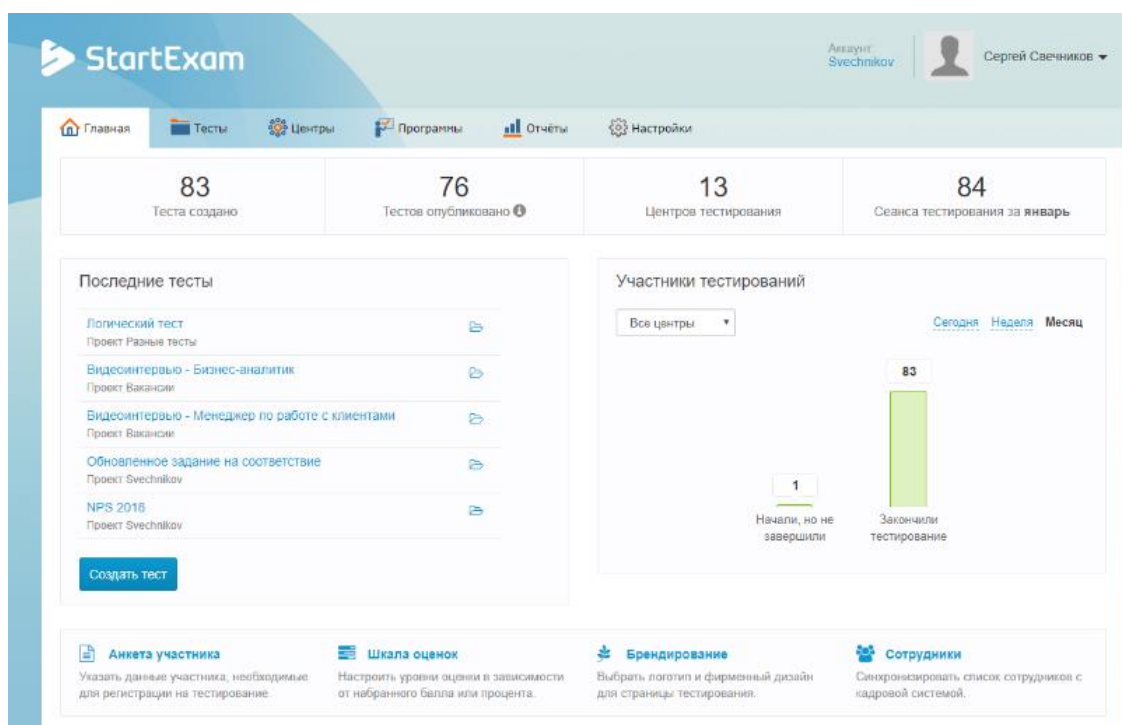


Рисунок 45 – Система тестирования StartExam

Для оценки конкурентоспособности разрабатываемой системы была составлена карта сравнения конкурентных технических решений, представленная в таблице 12. Индексом «ф» обозначена собственная разработка, индексом «к1» – система тестирования «INDIGO», индексом «к2» – система тестирования сотрудников «StartExam».

Таблица 12 – Карта сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда пользователя	0,08	4	5	4	0,30	0,38	0,30
2. Удобство в эксплуатации	0,08	3	4	3	0,23	0,30	0,23
3. Помехоустойчивость	0,05	3	4	3	0,14	0,18	0,14
4. Энергоэкономичность	0,03	5	1	5	0,15	0,03	0,15
5. Надёжность	0,06	4	5	4	0,24	0,30	0,24

Продолжение таблицы 12

6. Уровень шума	0,02	5	5	5	0,08	0,08	0,08
7. Безопасность	0,05	3	5	3	0,14	0,23	0,14
8. Потребность в ресурсах памяти	0,03	4	2	4	0,12	0,06	0,12
9. Функциональная мощность	0,06	3	5	4	0,18	0,30	0,24
10. Простота эксплуатации	0,06	5	2	4	0,30	0,12	0,24
11. Качество интеллектуального интерфейса	0,05	4	3	3	0,18	0,14	0,14
12. Возможность подключения к сети ЭВМ	0,02	5	5	5	0,08	0,08	0,08

Экономические критерии эффективности

1. Конкурентоспособность продукта	0,05	3	5	4	0,14	0,23	0,18
2. Уровень проникновения на рынок	0,08	1	4	3	0,08	0,30	0,23
3. Цена	0,06	5	3	4	0,30	0,18	0,24
4. Предполагаемый срок эксплуатации	0,03	4	5	3	0,12	0,15	0,09
5. Послепродажное обслуживание	0,06	3	5	5	0,18	0,30	0,30
6. Финансирование разработки	0,05	2	5	5	0,09	0,23	0,23
7. Срок выхода на рынок	0,05	3	2	4	0,14	0,09	0,18
8. Наличие сертификации разработки	0,08	1	5	1	0,08	0,38	0,08
Итого	1,00				3,26	4,06	3,62

Альтернативные решения для организации тестирований имеют более высокий коэффициент конкурентоспособности, чем разрабатываемый продукт. В новой следует сделать упор на ту часть функционала, которая является слабой у конкурентов.

Наиболее слабой стороной конкурентных решений является область их взаимодействия с пользователем. Система тестирования «INDIGO» сложна в первоначальной установке, а сервис «StartExam» в силу своей

ориентированности на проведение срезов знаний не содержит встроенных шаблонов психологических тестов.

Сильными сторонами разрабатываемой системы должны стать эргономичность пользовательского интерфейса и простота входа в неё новых пользователей.

Поскольку конкурентоспособность альтернативных решений высока для привлечения первичных клиентов стоимость реализации продукта стоит установить ниже, чем у конкурентов.

5.2.4 Оценка перспективности разработки

Для оценки перспективности разработки в соответствие с методологией QuaD была составлена оценочная карта, представленная в таблице 13. При составлении оценочной карты для каждого критерия экспертным был определён его вес таким образом, чтобы сумма всех весов равнялась единице.

Таблица 13 – Оценочная карта перспективности разработки

Критерий оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение
1	2	3	4	5	6
Показатели оценки качества разработки					
1. Энергоэффективность	0,08	80	100	0,8	0,06
2. Помехоустойчивость	0,08	60	100	0,6	0,05
3. Надёжность	0,05	60	100	0,6	0,03
4. Унифицированность	0,03	100	100	1	0,03
5. Уровень материалоемкости разработки	0,06	80	100	0,8	0,05
6. Уровень шума	0,02	100	100	1	0,02
7. Безопасность	0,05	60	100	0,6	0,03
8. Потребность в ресурсах памяти	0,03	80	100	0,8	0,02

Продолжение таблицы 13

9. Функциональная мощность	0,06	60	100	0,6	0,04
10. Простота эксплуатации	0,06	100	100	1	0,06
11. Качество интеллектуального интерфейса	0,05	80	100	0,8	0,04
12. Ремонтопригодность	0,02	100	100	1	0,02
Показатели оценки экономического потенциала разработки					
1. Конкурентоспособность продукта	0,05	60	100	0,6	0,03
2. Уровень проникновения на рынок	0,08	20	100	0,2	0,02
3. Перспективность рынка	0,06	100	100	1	0,06
4. Цена	0,03	80	100	0,8	0,02
5. Послепродажное обслуживание	0,06	60	100	0,6	0,04
6. Финансовая эффективность разработки	0,05	40	100	0,4	0,02
7. Срок выхода на рынок	0,05	60	100	0,6	0,03
8. Наличие сертификационной разработки	0,08	20	100	0,2	0,02
Итого	1				0,65

Полученное значение перспективности равно 65, из этого можно сделать вывод, что перспективность разработки выше среднего.

5.2.5 SWOT-анализ

SWOT-анализ – один из методов стратегического планирования, является простым и качественным инструментом оценивания конкурентоспособности. Суть метода заключается в выявлении ключевых факторов внутренней и внешней среды [8].

В таблице 14 приведён SWOT-анализ разрабатываемого решения.

Таблица 14 – SWOT-анализ технического решения

		Внутренние факторы	
		<p>Сильные стороны</p> <ol style="list-style-type: none"> 1. Удобный пользовательский интерфейс 2. Надёжная система авторизации 3. Низкий порог вхождения новых пользователей 4. Реализация в виде облачного сервиса 	<p>Слабые стороны</p> <ol style="list-style-type: none"> 1. Отсутствие государственной сертификации 2. Медленный старт из-за ограниченного рекламного бюджета 3. Наличие постоянных затрат на содержание облака. 4. Отсутствие квалифицированных сотрудников техподдержки
Внешние факторы	<p>Возможности</p> <ol style="list-style-type: none"> 1. Попадание продукта на первые строки в выдаче поисковых систем 2. Рост интереса к психологическим тестированиям 3. Получение гранта на развитие проекта 4. Помощь профессиональных психологов в создании тестов 5. Получение отзывов от пользователей 	<p>Сопоставление сильных сторон и возможностей говорит о том, что благодаря упрощённому интерфейсу у проекта есть возможность привлечь первичных клиентов, также есть шансы на дальнейшее увеличение их числа.</p>	<p>Исходя из слабых сторон и возможностей, можно предположить, что слабый уровень техподдержки может оттолкнуть покупателей и затормозить получение обратной связи о пожеланиях пользователей.</p>
	<p>Угрозы</p> <ol style="list-style-type: none"> 1. Усиление позиций конкурентов 2. Рост цен на использование облака 3. Снижение числа потенциальных клиентов в связи с ростом безработицы 	<p>Пересечение сильных сторон и угроз говорит о том, что проект слабо защищён от внешних угроз.</p>	<p>Использование внешнего облачного сервиса может подорвать бюджет проекта.</p>

	5. Введение дополнительных требований на сертификацию		
--	---	--	--

5.2.6 Определение возможных альтернатив разработки

Опираясь на то, как реализованы конкурентные решения, можно определить несколько альтернативных вариантов исполнения разработки:

1) Облачная веб-платформа без обязательной регистрации тестируемых сотрудников. Сотрудники получают доступ к тестированию по ссылке, выданной их руководителем. Личный кабинет имеет только руководитель предприятия.

2) Облачная веб-платформа с обязательным занесением сотрудников в базу данных. Личный кабинет есть как у руководителя предприятия, так и у сотрудника. Каждый сотрудник имеет доступ к личному кабинету с помощью логина и пароля, выданного его руководителем от предприятия.

3) Приложение для локального развёртывания внутри предприятия. Состоит из сервера и клиентских модулей. База данных сотрудников, а также авторизация может быть интегрирована с внутренней системой авторизации компании.

5.3 Планирование графика работ

5.3.1 Участники и структура работ

В реализации проекта принимает участие 3. В роли научного руководителя выступает Аксёнов Сергей Владимирович, в роли инженеров – Балыков Николай Ленскиевич и Рыбаченко Иван Александрович. Распределение работ по исполнителям приведено в таблице 15.

Таблица 15 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ работы	Содержание работ	Исполнитель
Разработка технического задания	1	Составление и утверждение технического задания	Аксёнов С.В.
Формирование концепции приложения	2	Определение компонентов системы, разграничение их ответственностей. Разработка диаграмм, описывающих внутреннее устройство системы.	Балыков Н.Л. Рыбаченко И.А.

	3	Выбор средств для разработки клиентской части	Рыбаченко И.А.
--	---	---	----------------

Продолжение таблицы 5

	4	Выбор средств для разработки серверной части	Балыков Н.Л.
Разработка дизайна приложения	5	Создание макетов страницы	Рыбаченко И.А.
	6	Создание логотипа	Рыбаченко И.А.
	7	Выбор цветовой схемы	Рыбаченко И.А.
Конструирование программного обеспечения	8	Реализация серверной части приложения	Балыков Н.Л.
	9	Конфигурирование базы данных	Балыков Н.Л.
	10	Реализация клиентской части приложения	Рыбаченко И.А.
Тестирование	11	Интеграционное тестирование приложения	Балыков Н.Л. Рыбаченко И.А.
	12	Пользовательское тестирование приложения	Аксёнов С.В. Балыков Н.Л. Рыбаченко И.А.
Оформление отчёта по НИР	13	Составление пояснительной записки к ВКР	Аксёнов С.В. Балыков Н.Л. Рыбаченко И.А.

5.3.2 Определение трудоёмкости выполнения работ

Для определения трудоёмкости работ была использована формула:

$$t_{exp} = \frac{3t_{min} + 2t_{max}}{5}, \quad (1)$$

где t_{exp} – ожидаемая трудоёмкость выполнения работы, чел.-дн.;

t_{min} – минимально возможная трудоёмкость выполнения заданной работы, чел.-дн.;

t_{max} – максимально возможная трудоёмкость выполнения заданной работы, чел.-дн.;

Продолжительность работ рассчитана по следующей формуле:

$$T_p = \frac{T_{exp}}{Ч}, \quad (2)$$

где T_p – продолжительность работы, дн.;

T_{exp} – ожидаемая трудоёмкость выполнения работы, чел.-дн.;

Ч – количество исполнителей, выполняющих одновременно одну и ту же работу

Для перевода продолжительности работы из рабочих дней в календарные используется следующая формула:

$$T_k = T_p \cdot k_{кал} \quad (3)$$

где T_k – продолжительность работы в календарных днях;

T_p – продолжительность работы в рабочих днях;

$k_{кал}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых}}, \quad (4)$$

где $k_{кал}$ – коэффициент календарности;

$T_{кал}$ – количество календарных дней в году;

$T_{вых}$ – количество выходных дней в году.

В 2020 году согласно производственному календарю [9] в расчёте на 6-дневную рабочую неделю содержится 366 календарных дней, 221 рабочий день и 145 выходных дней. Таким образом, согласно формуле 4, коэффициент календарности в 2020 году равен 1,66.

Результат расчётов трудоёмкости и продолжительности работ приведён в таблице 16.

Таблица 16 – Расчёт продолжительности выполнения работ

№ работы	t_{min}	t_{max}	t_{exp}	Количество исполнителей	T_p	T_k
1	1	7	4,6	1	4,6	8
2	7	28	19,6	2	9,8	17
3	1	7	4,6	1	4,6	8
4	1	7	4,6	1	4,6	8
5	7	28	19,6	1	19,6	33
6	1	7	4,6	1	4,6	8
7	1	7	4,6	1	4,6	8
8	28	72	54,4	1	54,4	91
9	7	28	19,6	1	19,6	33
10	28	56	44,8	1	44,8	75
11	1	7	4,6	2	2,3	4
12	1	7	4,6	3	1,6	3
13	14	28	22,4	3	7,5	13

5.3.3 Разработка графика проведения работ

Для визуализации продолжительности этапов разработки была построена диаграмма Ганта, представленная на рисунке 46.

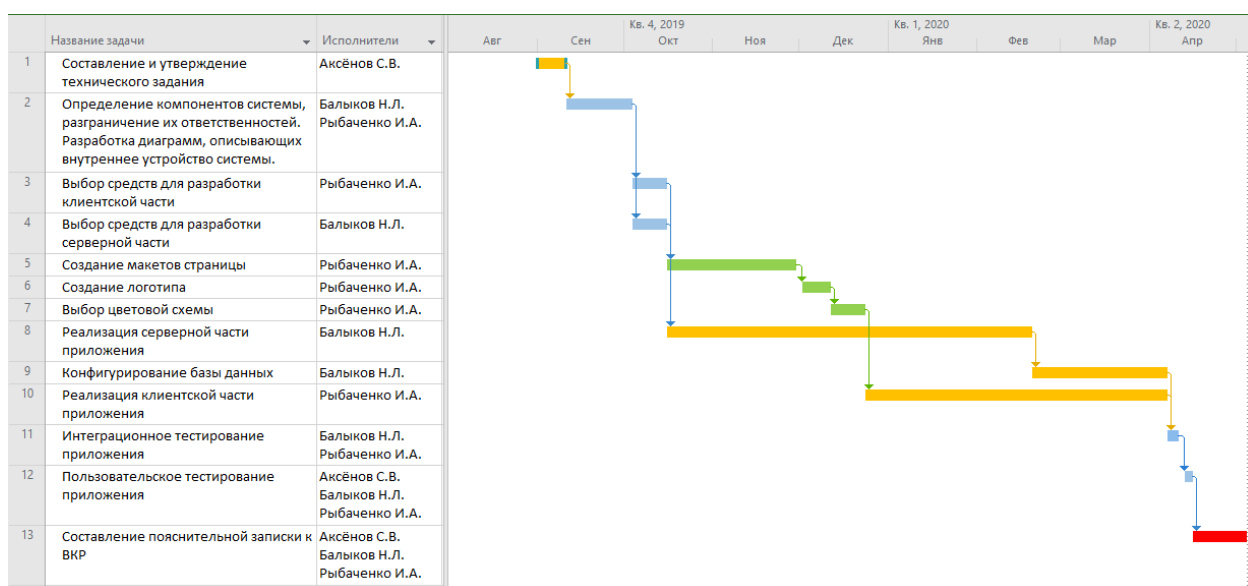


Рисунок 46 – Диаграмма Ганта

Общая продолжительность работ составила 8 месяцев. На рисунках 47, 48, 49 представлены диаграммы Ганта для отдельных этапов работ в более крупном масштабе.

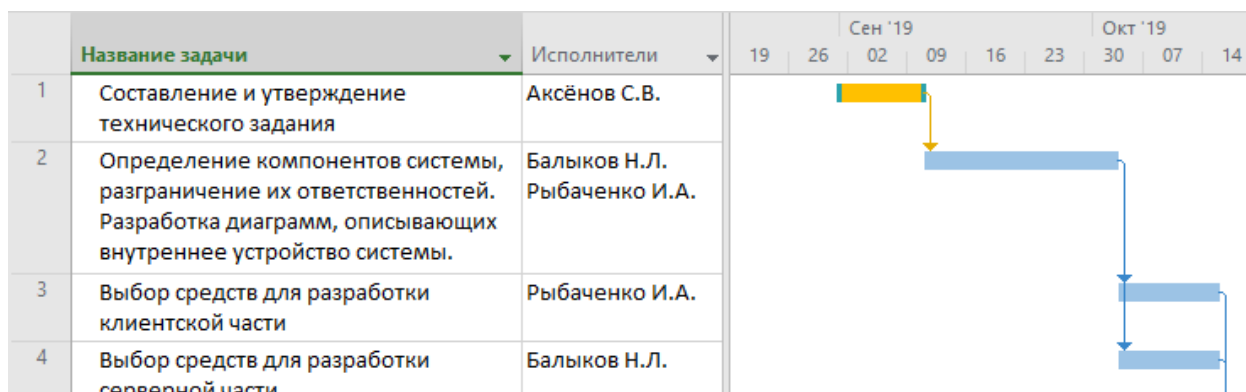


Рисунок 47 – Диаграмма Ганта. Этапы «Разработка технического задания» и «Формирование концепции приложения»

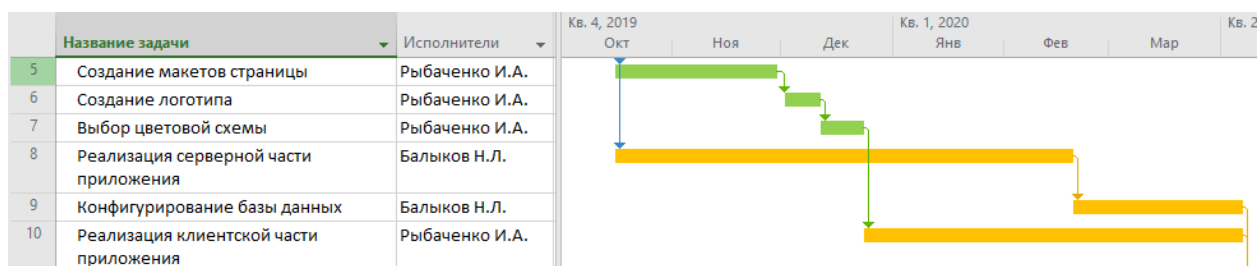


Рисунок 48 – Диаграмма Ганта. Этапы «Разработка дизайна приложения» и «Конструирование программного обеспечения»

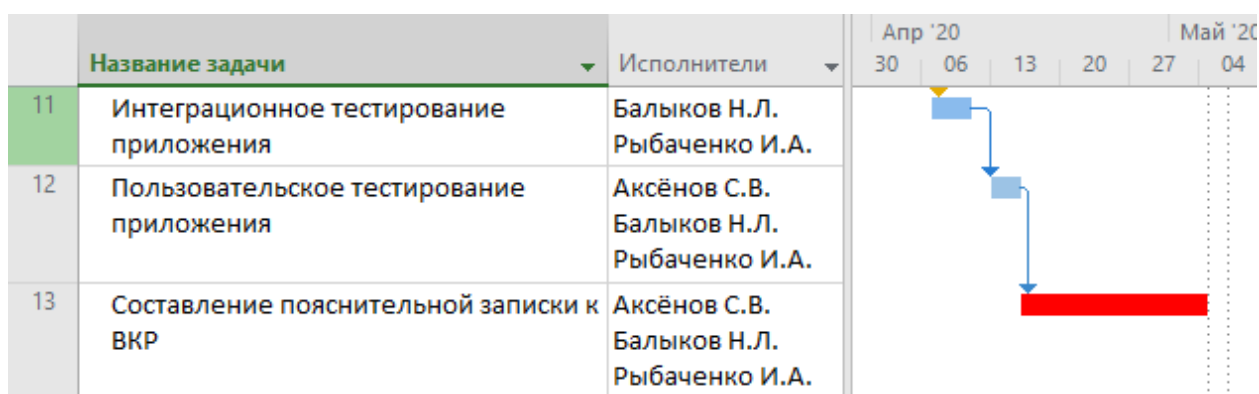


Рисунок 49 – Диаграмма Ганта. Этапы «Тестирование» и «Оформление отчёта по НИР»

5.4 Планирование и формирование бюджета научных исследований

5.4.1 Расчёт материальных затрат научно-технического исследования

При работе над исследовательской работой использовалось 2 компьютера средней мощности стоимость которых составляет 25000 и 35000 рублей. Срок полезного использования офисных машин (код 330.28.23.23) составляет от 2 до 3 лет. Для вычисления амортизации данный срок можно принять за 3 года. Материальные затраты, необходимые для научно-исследовательской работы сведены в таблицу 17.

Таблица 17 – Материальные затраты

Наименование	Единица измерения	Количество	Цена, руб.
Ноутбук Lenovo B50-45	шт.	1	25000
Ноутбук Asus	шт.	1	35000
Итого			60000

Норма амортизации вычисляется по следующей формуле:

$$A_n = \frac{100\%}{3} = 33,33\%$$

Годовые амортизационные вычисления составляют:

$$A_g = S \cdot \frac{A_n}{100\%} = 60000 \cdot 0,33 = 20000 \text{ рублей}$$

Ежемесячные амортизационные отчисления составят:

$$A_m = \frac{A_g}{12} = \frac{20000}{12} = 1666,66 \text{ рублей}$$

Итого за весь период выполнения исследовательской работы, с учётом того, что его продолжительность равна 8 месяцам, амортизация равна:

$$A = A_m \cdot 8 = 1666,66 \cdot 8 = 13333 \text{ рубля}$$

5.4.2 Основная заработная плата исполнителей исследовательской работы

Зарплата исполнителей складывается из двух частей – основной и дополнительной и может быть выражена следующей формулой:

$$Z_{\text{п}} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (5)$$

где $Z_{\text{п}}$ – полная зарплата сотрудника;

$Z_{\text{осн}}$ – основная зарплата сотрудника;

$Z_{\text{доп}}$ – дополнительная зарплата сотрудника.

Основная зарплата – это та часть зарплаты, которая выплачивается работнику гарантированно, не зависимо от результатов труда. Для расчёта основной зарплаты используется формула:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_p, \quad (6)$$

где $Z_{\text{осн}}$ – основная зарплата работника;

$Z_{\text{дн}}$ – среднедневная заработная плата работника;

T_p – продолжительность работы в днях.

Среднедневная заработная плата определяется формулой:

$$Z_{\text{дн}} = \frac{Z_m \cdot M}{F_d}, \quad (7)$$

где $Z_{\text{дн}}$ – среднедневная зарплата работника;

Z_m – месячный должностной оклад работника;

M – количество месяцев работы без отпуска за год (10,4 мес.);

F_d – действительный годовой фонд рабочего времени сотрудника.

Для определения месячного должностного оклада работника используется следующая формула:

$$Z_m = Z_{тс} \cdot (1 + k_{пр} + k_d) \cdot k_p, \quad (8)$$

где Z_m – месячный должностной оклад работника;

$Z_{тс}$ – заработная плата по тарифной ставке;

$k_{пр}$ – премиальный коэффициент, равный 0,3;

k_d – коэффициент надбавок;

k_p – районный коэффициент, для Томска равный 1,3.

Расчёт баланса рабочего времени приведён в таблице 18.

Таблица 18 – Расчёт баланса трудового времени

Показатель рабочего времени	Дни
Календарные дни	366
Нерабочие дни (выходные\праздники)	145
Потери рабочего времени (отпуск\невыходы по болезни)	56
Действительный годовой фонд рабочего времени	165

Результат расчёта основной заработной платы приведён в таблице 19. Зарплата студента принимается за 20000 рублей в месяц, а зарплата преподавателя за 30000 рублей.

Таблица 19 – Расчёт основной заработной платы работников

Исполнитель	$Z_{т,}$ руб	$k_{пр}$	$k_{д}$	$k_{р}$	$Z_{м,}$ руб	$Z_{дн,}$ руб	$T_{р,}$ дней	$Z_{осн,}$ руб
Аксёнов Сергей Владимирович	60000	0,3	0,2	1,3	30000	1890,91	24	45381,84
Балыков Николай Ленскиевич	30000	0,3	0,2	1,3	20000	1260,61	169	213043,09
Рыбаченко Иван Александрович	30000	0,3	0,2	1,3	20000	1260,61	169	213043,09
Итого								471468,02

Общий размер основной заработной платы составил 471468,02 рублей.

5.4.3 Дополнительная заработная плата

Дополнительная зарплата назначается за совмещение работы с учёбой, при предоставлении ежегодного оплачиваемого отпуска и пр. Дополнительная заработная плата рассчитывается умножением на надбавочный коэффициент. Величина надбавочного коэффициента в рамках научной работы была принята за 0,15.

Результат расчёта дополнительной заработной платы работников приведён в таблице 20.

Таблица 20 – Расчёт дополнительной заработной платы работников

Исполнитель	Основная заработная плата	Надбавочный коэффициент	Дополнительная заработная плата
Аксёнов Сергей Владимирович	45381,84	0,15	6807,28
Балыков Николай Ленскиевич	213043,09		31956,46
Рыбаченко Иван Александрович	213043,09		31956,46
Итого			70720,2

Общий размер дополнительной заработной платы составил 70720,2 рублей.

5.4.4 Отчисления во внебюджетные фонды

Данная статья включает обязательные отчисления в фонд социального страхования, пенсионный фонд и фонд медицинского страхования. Размер отчислений зависит от размера заработной платы по следующей формуле:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (9)$$

где $Z_{\text{внеб}}$ – размер отчислений на уплату во внебюджетные фонды;

$k_{\text{внеб}}$ – коэффициент отчислений во внебюджетные фонды;

$Z_{\text{осн}}$ – основная заработная плата работника;

$Z_{\text{доп}}$ – дополнительная заработная плата работника.

В таблице 21 представлен результат расчёта отчислений во внебюджетные фонды.

Таблица 21 – Расчёт отчислений во внебюджетные фонды

Исполнитель	Основная заработная плата	Дополнительная заработная плата	Коэффициент отчислений во внебюджетные фонды	Сумма отчислений во внебюджетные фонды
Аксёнов Сергей Владимирович	45381,84	6807,28	0,271	14143,26
Балыков Николай Ленскиевич	213043,09	31956,46		66394,88
Рыбаченко Иван Александрович	213043,09	31956,46		66394,88
Итого				146933,02

Суммарный размер отчислений во внебюджетные фонды составил 146933,02 рублей.

5.4.5 Накладные расходы

Накладные расходы включают в себя затраты, не попавшие в ранее описанные статьи. К таким расходам относятся оплата электроэнергии, услуг связи, ксерокопирование и прочие услуги. Величина накладных расходов определяется из формулы:

$$Z_{\text{накл}} = k_{\text{нр}} \cdot \sum_{i=1}^4 \text{Сумма } i \text{ – ой статьи}, \quad (10)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы (16%).

Согласно вышеприведённой формуле и ранее выполненным расчётам размер накладных расходов составляет 111040,15 рублей.

5.4.6 Бюджет затрат научно-исследовательского проекта

В результате сбора всех статей расходов в одну таблицу получается бюджет исследования. Результат формирования бюджета представлен в таблице 22.

Таблица 22 – Бюджет затрат научно-исследовательского проекта

Наименование статьи	Сумма, руб.	Примечание
1. Амортизационные затраты на оборудование	13333	См. раздел 1.4.1
2. Затраты на основную заработную плату	471468,02	См. раздел 1.4.2
3. Затраты на дополнительную заработную плату	70720,2	См. раздел 1.4.3
4. Отчисления во внебюджетные фонды	146933,02	См. раздел 1.4.4
5. Накладные расходы	111040,15	См. раздел 1.4.5
Итого	799380,64	Сумма статей 1-5

Таким образом, бюджет разработки составляет 800 тысяч рублей.

5.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Для определения интегрального финансового показателя используется формула:

$$I_{\text{финр}}^{\text{исп } i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (11)$$

где $I_{\text{финр}}^{\text{исп } i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

Для определения интегрального показателя ресурсоэффективности используется формула:

$$I_{pi} = \sum a_i b_i, \quad (12)$$

где I_{pi} – интегральный показатель ресурсоэффективности;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки.

При расчёте показателей эффективности оценивались 3 варианта исполнения системы:

1) Облачная веб-платформа без обязательной регистрации тестируемых сотрудников. Сотрудники получают доступ к тестированию по ссылке, выданной их руководителем. Личный кабинет имеет только руководитель предприятия.

2) Облачная веб-платформа с обязательным занесением сотрудников в базу данных. Личный кабинет есть как у руководителя предприятия, так и у

сотрудника. Каждый сотрудник имеет доступ к личному кабинету с помощью логина и пароля, выданного его руководителем от предприятия.

3) Приложение для локального развёртывания внутри предприятия. Состоит из сервера и клиентских модулей. База данных сотрудников, а также авторизация может быть интегрирована с внутренней системой авторизации компании.

Расчёт интегрального показателя ресурсоэффективности для описанных выше вариантов исполнения приведён в таблице 23.

Таблица 23 – Расчёт интегральных показателей ресурсоэффективности

Параметр оценки	Весовой коэффициент параметра	Исполнение 1	Исполнение 2	Исполнение 3
Способствует росту производительности труда	0,15	5	4	4
Удобство в эксплуатации	0,25	5	4	5
Помехоустойчивость	0,10	4	4	5
Энергосбережение	0,15	4	4	2
Надёжность	0,25	4	3	5
Материалоёмкость	0,10	4	3	2
Интегральный коэффициент ресурсоэффективности		4,4	3,65	4,1

Интегральный показатель эффективности варианта исполнения определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{испi} = \frac{I_{р-испi}}{I_{финр}^{испi}}, \quad (13)$$

где $I_{испi}$ – интегральный показатель эффективности варианта исполнения;

$I_{р-испi}$ – интегральный показатель ресурсоэффективности;

$I_{финр}^{испi}$ – интегральный финансовый показатель.

Сравнительная эффективность варианта исполнения проекта определяется формулой:

$$\mathcal{E}_{ср} = \frac{I_{испi}}{I_{испо}}, \quad (14)$$

где $\mathcal{E}_{ср}$ – сравнительная эффективность варианта исполнения проекта;

$I_{испi}$ – интегральный показатель эффективности варианта исполнения проекта;

$I_{испо}$ – интегральный показатель эффективности лучшего варианта исполнения проекта.

В таблице 24 представлен результат расчёта показателей эффективности для тех же трёх вариантов исполнения.

Таблица 24 – Показатели эффективности разработки

№	Показатель	Исполнение 1	Исполнение 2	Исполнение 3
1	Интегральный финансовый показатель разработки	0,9	0,95	1
2	Интегральный показатель ресурсоэффективности разработки	4,4	3,65	4,1
3	Интегральный показатель эффективности	4,9	3,8	4,1
4	Сравнительная эффективность вариантов исполнения	1	0,79	0,84

5.6 Вывод по главе

В разделе исследовательской работы, посвященному финансовому менеджменту, ресурсоэффективности и ресурсосбережению, была дана оценка коммерческого потенциала разработки, спланирован график работ, сформирован бюджет затрат и определена эффективность исследования.

Для оценки коммерческого потенциала использовались методика Quad и метод SWOT-анализа. В результате оценки сделан вывод о том, что перспективность разработки выше среднего, а конкурентоспособность разработки ниже, чем у конкурентов. Наиболее сильной стороной разработки является простой пользовательский интерфейс, а наиболее слабой – зависимость от стоимости облачных вычислительных ресурсов.

При планировании графика работ был составлен список задач, для каждой из которых определены исполнители и продолжительность. График работ визуализирован в виде диаграммы Ганта. С учётом продолжительности работ сформирован бюджет затрат научного исследования, размер которого составил 800 тысяч рублей.

Сравнение интегральных показателей эффективности вариантов исполнения показало, что наиболее выгодным с точки зрения ресурсоэффективности является 1 вариант исполнения, который и был реализован.

Глава 6. Социальная ответственность

6.1 Введение

В рамках выпускной квалификационной работы была создана платформа психологического тестирования для корпоративных клиентов. Проведение психологических тестирований позволяет своевременно оценить уровень удовлетворенности работой сотрудников организации, а также выявить факторы, негативно влияющие на психологический климат внутри трудового коллектива.

Целью работы является формирование благоприятного психологического климата в трудовом коллективе и повышение заинтересованности персонала в работе.

Разработанная платформа психологического тестирования для корпоративных клиентов представляет собой интернет-приложение, автоматизирующее процессы, связанные с проведением ручного тестирования членов организаций. Зарегистрировавшись на платформе психологического тестирования, представитель организации может выбрать подходящий тест и назначить дату проведения тестирования для сотрудников. Затем система генерирует уникальную ссылку на тестирование и предоставляет её представителю организации. Сотрудники организации получают ссылку от представителя организации и проходят тестирование, указав своё ФИО и при необходимости возраст, должность, отдел и другие личные данные.

Проектирование и разработка платформы психологического тестирования для корпоративных клиентов велась в Томском политехническом университете, место проведения работ – Кибернетический центр. В процессе работы были использованы персональный компьютер и ноутбук, в связи с чем авторы могли подвергнуться различным вредным факторам, рассмотренным далее.

6.2 Правовые и организационные вопросы обеспечения безопасности

6.2.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.

Трудовые отношения между работодателем и работником регулируются с помощью законодательного акта "Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020), что позволяет урегулировать вопросы, связанные с организацией труда, управлением трудом, заработной платой, трудовыми спорами и другие [10]. Ниже приведены его наиболее важные для соблюдения фрагменты:

- Нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю. Порядок исчисления нормы рабочего времени на определенные календарные периоды (месяц, квартал, год) в зависимости от установленной продолжительности рабочего времени в неделю определяется федеральным органом исполнительной власти, осуществляющим функции по выработке государственной политики и нормативно-правовому регулированию в сфере труда.
- В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов (в ред. Федерального закона от 18.06.2017 N 125-ФЗ).
- Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

6.2.2 Организационные мероприятия при компоновке рабочей зоны.

Разработка платформы психологического тестирования для корпоративных клиентов велась в помещении общей площадью 26 квадратных метров, в качестве искусственного источника освещения использовались 3 лампы накаливания, общей мощностью 85 Вт. Помещение оборудовано одним крупным столом, позволяющим работать одновременно группе из двух человек, и одним компьютерным столом стандартного размера с выдвижной клавиатурой. Участники научно-исследовательской группы по разработке платформы психологического тестирования для корпоративных клиентов взаимодействовали с электронно-вычислительными устройствами, находясь в операторских креслах, выполненных в виде компьютерного кресла с регулируемыми подлокотниками и углом наклона спинки. Перемещение кресла внутри помещения обеспечивают 5 пластиковых колес диаметром 50 мм. Каждый участник научно-исследовательской группы по разработке платформы психологического тестирования для корпоративных клиентов постоянно имел доступ к свободному операторскому креслу.

С целью минимизации воздействия вредных факторов на членов научно-исследовательской группы по разработке платформы психологического тестирования для корпоративных клиентов при выполнении научно-исследовательской работы, рабочее место должно быть организовано с учетом требований ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» и СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Ниже приведены наиболее важные для соблюдения фрагменты стандарта ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [11]:

1. Подвижность кресла относительно пола или другой поверхности, на которой оно установлено, может не ограничиваться. В случае необходимости обеспечения строго определенного положения человека-оператора по отношению к средствам отображения информации и органам управления, а также в случае, если трудовая деятельность человека-оператора сопряжена с силовыми и резкими движениями, кресло должно быть фиксировано. При этом, в зависимости от характера трудовой деятельности оператора, должна быть обеспечена возможность изменения положения кресла или сиденья в горизонтальной плоскости с фиксацией его в нужном положении. При необходимости подвижность кресла должна задаваться также вращением кресла на 180-360° вокруг вертикальной оси опорной конструкции кресла с фиксацией в нужном положении.

2. При работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.

3. Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

4. В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

5. Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

6. В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях

допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

7. Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Проектной группой научно-исследовательской работы по разработке платформы психологического тестирования для корпоративных клиентов были соблюдены в допустимой мере все требования, предусматриваемые государственным стандартом 12.2.032-78. Во время выполнения выпускной квалификационной работы не происходило случаев, несущих в себе угрозу для здоровья и жизни сотрудников и участников проектной группы, а также предоставления опасности для окружающей среды.

6.2.3 Особенности законодательного регулирования проектных решений

На основании федерального закона Российской Федерации «О персональных данных», №152-ФЗ, глава 2, статья 5 [12]:

1. Обработка персональных данных должна осуществляться на законной и справедливой основе.

2. Обработка персональных данных должна ограничиваться достижением конкретных, заранее определенных и законных целей. Не допускается обработка персональных данных, несовместимая с целями сбора персональных данных.

3. Не допускается объединение баз данных, содержащих персональные данные, обработка которых осуществляется в целях, несовместимых между собой.

4. Обработке подлежат только персональные данные, которые отвечают целям их обработки.

5. Содержание и объем обрабатываемых персональных данных должны соответствовать заявленным целям обработки. Обрабатываемые персональные данные не должны быть избыточными по отношению к заявленным целям их обработки.

6. При обработке персональных данных должны быть обеспечены точность персональных данных, их достаточность, а в необходимых случаях и актуальность по отношению к целям обработки персональных данных. Оператор должен принимать необходимые меры либо обеспечивать их принятие по удалению или уточнению неполных, или неточных данных.

7. Хранение персональных данных должно осуществляться в форме, позволяющей определить субъекта персональных данных, не дольше, чем этого требуют цели обработки персональных данных, если срок хранения персональных данных не установлен федеральным законом, договором, стороной которого, выгодоприобретателем или поручителем, по которому является субъект персональных данных. Обрабатываемые персональные данные подлежат уничтожению либо обезличиванию по достижении целей обработки или в случае утраты необходимости в достижении этих целей, если иное не предусмотрено федеральным законом.

Исходя из положения федерального закона, участниками научно-исследовательской группы по разработке платформы психологического тестирования для корпоративных клиентов было принято решение добавить в клиентские компоненты интерфейса, связанные с указанием персональных данных, графические элементы, сообщающие пользователям о согласии на обработку персональной информации.

6.3 Производственная безопасность

В данном подразделе производится анализ вредных и опасных факторов, которые могут возникнуть на одном из этапов выполнения работ по разработке платформы психологического тестирования для корпоративных клиентов.

Отказ от рассмотрения химических и шумовых факторов обусловлен незначительностью их наличия на рабочем месте или их полным отсутствием.

Таблица 25 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Проектирование	Разработка	Эксплуатация	
1.Отклонение показателей микроклимата	+	+	+	1.СанПиН 2.2.4548-96 [13] 2.СП 52.13330.2010[14] 3.СанПиН 2.2.1/2.1.1.1278–03 [15] 4.СанПиН 2.1.8/2.2.4.1383-03 [16]
2.Отсутствие или недостаток естественного света	+	+	+	
3.Недостаточная освещенность рабочей зоны	+	+	+	
4.Повышенный уровень электромагнитных излучений	+	+	+	

Как видно из таблицы 25, опасные и вредные факторы воздействовали на участников проектной группы по разработке платформы психологического тестирования для корпоративных клиентов на протяжении всех работ, так как все этапы научно-исследовательской работы были произведены в одном и том же помещении.

Также стоит отметить, что воздействию вредных факторов подвержены потенциальные пользователи платформы психологического тестирования для корпоративных клиентов.

6.3.1 Анализ опасных и вредных производственных факторов

Представленные выше производственные факторы оказывают негативное влияние на организм человека, поэтому необходимо знать о допустимых нормах их воздействия. Данные о влиянии и допустимых нормах опасных и вредных факторов представлены в таблице 26.

Таблица 26 – Влияние опасных и вредных факторов

Фактор	Источник	Воздействие	Допустимые нормы
Отклонение показателей микроклимата	Отсутствие кондиционеров	Вялость, усталость, сниженная концентрация	Таблица 3
Отсутствие или недостаток естественного света	Периодическая необходимость работы за ЭВМ в ночное время	Ухудшение зрения, усталость глаз	КЕО не ниже 1,2 %-1,5 %
Недостаточная освещенность рабочей зоны	Недостаточная мощность осветительных приборов	Ухудшение зрения, усталость глаз	Освещенность на рабочей поверхности от системы общего искусственного освещения 200-300 лк.
Повышенный уровень электромагнитных излучений	Компоненты персональных компьютеров и ноутбуков	Возможно возникновение рака	Напряженность электростатического поля не более 20 кВ/м

Норма микроклимата является плавающим параметром и зависит от температуры помещения, поверхностей, влажности и скорости воздуха [17]. Допустимые величины показателей микроклимата продемонстрированы в таблице 27.

Таблица 27 – Допустимые величины показателей микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	20,0-21,9	19,0-26,0	15-75	0,1
Теплый	21,0-22,9	20,0-29,0		

6.3.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов

Для восстановления и поддержания допустимого микроклимата необходимо придерживаться следующих правил:

- Оборудование помещения системами обогрева, вентилирования и увлажнения
- Защита фасада здания от солнца: шторы, жалюзи, навесы и т.д.
- Рационально размещать рабочие места
- Своевременная влажная уборка помещения

Для решения проблемы отсутствия или недостатка естественного света и плохой освещенности рабочего места подходят следующие пункты:

- Своевременная чистка стекол в светопроемах
- Ремонт помещения в светлых тонах
- Установка более мощных ламп или в большем количестве в правильном положении

Повышенный уровень электромагнитных излучений можно избежать, если следовать следующим пунктам:

- Использовать монитор с жидкокристаллическим экраном

- Располагать монитор в углу помещения для того, чтобы стены поглощали излучение
- Выключать компьютер при его неиспользовании
- Сокращать время, проводимое за компьютером

6.4 Экологическая безопасность

Для выполнения научно-исследовательской работы по разработке платформы психологического тестирования для корпоративных клиентов использовались персональные компьютеры и ноутбуки. Современные электронно-вычислительные устройства не выбрасывают в окружающую среду каких-либо вредных веществ, однако используют для работы электроэнергию и создают электромагнитные поля. Рост потребления электроэнергии может привести к строительству дополнительных электростанций, что увеличивает общий уровень вредных выбросов и повышает вероятность возникновения экологической катастрофы. Производство и утилизация современных вычислительных устройств составляют серьезную проблему: текстолит, используемый при производстве микросхем, имеющих срок разложения более тысячи лет.

Мероприятия, позволяющие сохранять экологическую безопасность находясь на своем рабочем месте:

- Правильная утилизация персональных компьютеров и ноутбуков, а также их комплектующих
- Использование энергосберегающих ламп
- Использование аккумуляторов вместо солевых батареек
- Своевременное отключение неиспользуемых электроприборов

6.5 Безопасность в чрезвычайных ситуациях

6.5.1 Пожар

Научно-исследовательская работа по разработке платформы психологического тестирования для корпоративных клиентов проходила в офисе. Одной из наиболее возможных чрезвычайных ситуаций в помещениях такого типа является пожар. К пожару могут привести неисправности в технических средствах, оргтехнике, а также действия самих сотрудников [18]. Главное во время пожара – не поддаваться панике и действовать согласно правилам поведения в подобной чрезвычайной ситуации:

1. Заметив пожар или загорание, необходимо немедленно организовать оповещение об этом всех находящихся в здании людей, независимо от размеров и места пожара или загорания, равно как и при обнаружении хотя бы малейших признаков горения (дыма, запаха гари) и немедленно вызвать пожарную охрану по телефону «01».
2. Сообщения о пожаре, как правило, передаются по телефону. Поэтому каждый человек должен хорошо знать места расположения телефонных аппаратов, особенно тех, которые доступны в любое время суток.
3. Каждый работник образовательного учреждения, обнаруживший пожар или его признаки (задымление, запах горения или тления различных материалов, повышение температуры и т.п.) обязан:
4. немедленно сообщить об этом по телефону в пожарную часть (при этом необходимо четко назвать адрес учреждения, место возникновения пожара, а также сообщить свою должность, фамилию и номер своего телефона);
5. задействовать систему оповещения людей о пожаре, приступить самому и привлечь других лиц к эвакуации детей из здания в безопасное место согласно плану эвакуации;

6. известить о пожаре руководителя образовательного учреждения или заменяющего его работника;
7. организовать встречу пожарных подразделений, принять меры по тушению пожара имеющимися в учреждении средствами пожаротушения.

6.6 Вывод по разделу

Подводя итоги по данному разделу, можно отметить, что проектная деятельность по разработке платформы психологического тестирования для корпоративных клиентов соответствовала всем заявленным нормам безопасности жизнедеятельности. Рабочее место во время проведения научно-исследовательской работы соответствовало указанным стандартам, а также санитарно-эпидемиологическим правилам и нормам. Особенно стоит отметить соблюдение федерального закона «О персональных данных» и готовность участников проектной группы к чрезвычайным ситуациям.

Заключение

Проведение психологических тестирований позволяет оценивать ситуацию в трудовом коллективе и, как следствие, формировать благоприятный психологический климат в коллективе и способствовать росту заинтересованности персонала в работе.

Основные результаты проведенной работы:

- Проведен обзор на тему корпоративных психологических тестирований и программных средств проведения корпоративного психологического тестирования.
- Спроектирована платформа психологического тестирования для корпоративных клиентов.
- Выполнена программная реализация платформы психологического тестирования для корпоративных клиентов.
- Описаны результаты проделанной работы.

В результате работы было разработано веб-приложение для проведения психологических тестирований. При разработке использовались такие языки программирования как Java, JavaScript, TypeScript.

Разработанная платформа была рекомендована руководителем разработки к использованию на факультете психологии ТГУ для оценки удовлетворенности учащихся колледжей процессом обучения.

Список литературы

1. Тестирование. Тесты, их классификация. [Электронный ресурс]. – Режим доступа: <https://www.psychologos.ru/articles/view/testirovanie>, свободный (дата обращения: 11.05.2020).
2. Психологические тесты онлайн. [Электронный ресурс]. – Режим доступа: <https://psyttests.org>, свободный (дата обращения: 12.05.2020).
3. Программа для создания тестов и онлайн тестирования. [Электронный ресурс]. – Режим доступа: <https://indigotech.ru>, свободный (дата обращения: 12.05.2020).
4. Система тестирования сотрудников. [Электронный ресурс]. – Режим доступа: <https://www.startexam.ru>, свободный (дата обращения: 12.05.2020).
5. REST. Общие принципы организации взаимодействия приложения/сайта с сервером через протокол HTTP [Электронный ресурс]. – Режим доступа: <https://dashvlas.com/blog/rest>, свободный (дата обращения: 14.05.2020).
6. Внедрение компонентного подхода в вебе: обзор веб-компонентов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/microsoft/blog/264791>, свободный (дата обращения: 14.05.2020).
7. Мутации [Электронный ресурс]. – Режим доступа: <https://vuex.vuejs.org/ru/guide/mutations.html>, свободный (дата обращения: 14.05.2020).
8. SWOT-анализ [Электронный ресурс]. – Режим доступа: <https://www.calltouch.ru/glossary/swot-analiz/>, свободный (дата обращения: 16.05.2020).

9. Производственный календарь 2020 года [Электронный ресурс]. – Режим доступа: <http://www.calendar.yoip.ru/work/2020-proizvodstvennyj-calendar.html>, свободный (дата обращения: 16.05.2020).
10. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 24.04.2020) [Электронный ресурс]. – Режим доступа: <https://ppt.ru/kodeks/tk>, свободный (дата обращения: 20.05.2020).
11. ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200003913> (дата обращения: 20.05.2020).
12. Федеральный закон Российской Федерации «О персональных данных» [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_61801 (дата обращения: 20.05.2020).
13. СанПиН 2.2.4548-96 «Гигиенические требования к микроклимату производственных помещений» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/901704046> (дата обращения: 20.05.2020).
14. СП 52.13330.2010 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200084092> (дата обращения: 20.05.2020).
15. СанПиН 2.2.1/2.1.1.1278–03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/901859404> (дата обращения: 20.05.2020).
16. СанПиН 2.1.8/2.2.4.1383-03 «Гигиенические требования к размещению и эксплуатации передающих радиотехнических объектов» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/901865556> (дата обращения: 20.05.2020).

17. Оптимальные и допустимые параметры воздуха [Электронный ресурс]. – Режим доступа: <http://www.technoholod.ru/site.aspx?SECTIONID=1940097> (дата обращения: 20.05.2020).

18. Должностные инструкции при пожаре [Электронный ресурс]. – Режим доступа: <https://hr-portal.ru/doki/dolzhnostnaya-instrukciya-specialista-po-rozharnoy-bezopasnosti> (дата обращения: 21.05.2020).

Приложение А. Описание классов бэкенд-сервера

Ниже приведена таблица, содержащая описание классов бэкенд-сервера.

Таблица А.1 – Описание классов бэкенд-сервера

Имя класса	Назначение
TestController	Класс-контроллер тестов. Принимает и обрабатывает запросы по маппингу /test
TestingController	Класс-контроллер тестирований. Принимает и обрабатывает запросы по маппингу /testing
Company	Класс сущности компания. Описывает бизнес-сущность компаний
CompanyType	Набор возможных типов компаний. Основные типы: Образовательное учреждение, Государственное учреждение, Частная организация
PossibleAnswer	Класс сущности вариант ответа. Описывает бизнес-сущность варианта ответа на вопрос в тесте
Profile	Класс сущности профиль. Описывает сущность пользователя веб-приложения
ProfileType	Набор возможных типов профилей. Основные типы: Администратор, Представитель организации
Question	Класс сущности вопрос. Описывает бизнес-сущность вопроса в тесте

Test	Класс сущности тест. Описывает бизнес-сущность теста
Testing	Класс сущности тестирование. Описывает бизнес-сущность тестирование
TestingResult	Класс сущности результат тестирования. Описывает бизнес-сущность результата тестирования
TestingStatus	Набор возможных типов статуса тестирования. Основные типы: В процессе, Завершен
TestingRepository	Класс-репозиторий тестирований. Осуществляет взаимодействие с таблицей Тестирования и её связями в БД
TestRepository	Класс-репозиторий тестов. Осуществляет взаимодействие с таблицей Тест и её связями в БД
TestingResultRepository	Класс-репозиторий результатов тестирований. Осуществляет взаимодействие с таблицей РезультатТестирования и её связями в БД
TestingService	Интерфейс сервиса тестирований. Указывает методы управления бизнес-сущностью Тестирование
TestingServiceImpl	Класс-сервис тестирований. Реализует методы интерфейса сервиса тестирований

TestService	Интерфейс сервиса тестов. Указывает методы управления бизнес-сущностью Тест
TestServiceImpl	Класс-сервис тестов. Реализует методы интерфейса сервиса тестов
TestingPlatformApplication	Класс, содержащий метод main. Является точкой входа приложения.
ExcelService	Интерфейс сервиса отчетов. Указывает методы для генерации отчета о тестировании в Excel.
ExcelServiceImpl	Класс-сервис отчетов. Реализует методы интерфейса сервиса отчетов.

Приложение Б. Программный код контроллера тестов

Контроллер тестов предоставляет CRUD интерфейс тестов для администратора платформы психологического тестирования для корпоративных клиентов.

Листинг Б.1 – Программный код контроллера тестов

```
/**
 * Контроллер тестов
 */
@RestController
@RequestMapping("test")
public class TestController {
    private final TestService testService;

    public TestController(TestService testService) {
        this.testService = testService;
    }

    @GetMapping
    public List<Test> getAllTests() {
        return testService.getAllTests();
    }

    @GetMapping("/{id}")
    public Test getTestById(@PathVariable("id") Test test) {
        return test;
    }

    @PostMapping
    public Test createTest(@RequestBody Test test) {
        return testService.saveTest(test);
    }

    @PutMapping("/{id}")
    public Test updateTest(@PathVariable("id") Test testFromDb,
        @RequestBody Test test) {
        BeanUtils.copyProperties(test, testFromDb, "id");
        return testService.saveTest(testFromDb);
    }

    @DeleteMapping("/{id}")
    public void deleteTest(@PathVariable("id") Test test) {
        testService.deleteTest(test);
    }
}
```