

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных
и гуманитарных дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Папина
подпись
« ____ » _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка информационной системы профориентационного тестирования
для ХТИ – филиала СФУ

Руководитель _____ доцент, канд. физ-мат. наук М. А. Буреева
подпись, дата

Выпускник _____ О. А. Доронина
подпись, дата

Консультанты
по разделам:

Экономический _____ М. А. Буреева
подпись, дата

Нормоконтролер _____ А. Н. Кадычегова
подпись, дата

Абакан 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных
и гуманитарных дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Папина
подпись
« ____ » _____ 2023 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Абакан 2023

Студенту Дорониной Олеся Анатольевне

Группа ХБ 19-02

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка информационной системы профориентационного тестирования для ХТИ – филиала СФУ

Утверждена приказом по институту № 283 от 11.05.2023 г.

Руководитель ВКР: М. А. Буреева, доцент, канд. физ-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области.
2. Описание процесса разработки информационной системы профориентационного тестирования.
3. Оценка экономической эффективности разработки и внедрения информационной системы для профориентационного тестирования.

Перечень графического материала: нет

Руководитель ВКР

подпись

М. А. Буреева

Задание принял к исполнению

подпись

О. А. Доронина

«11» мая 2023г.

РЕФЕРАТ

Выпускная квалификационная работа (ВКР) по теме «Разработка информационной системы профориентационного тестирования для ХТИ – филиала СФУ» содержит 90 страниц текстового документа, 10 таблиц, 46 иллюстраций, 8 формул, 18 использованных источников, 2 приложения.

ПРОФОРИЕНТАЦИОННОЕ ТЕСТИРОВАНИЕ, ВЕБ-ПРИЛОЖЕНИЕ, ИНФОРМАЦИОННАЯ СИСТЕМА, РЕСПОНДЕНТ, ОДПИНН, ЦПЮИ, СПЕЦИАЛИСТ ОДПИНН, IDEF3, DFD, PYTHON, FLASK.

Объект ВКР: методы автоматизации процессов тестирования, анкетирования, опросов.

Предмет ВКР: информационная система профориентационного тестирования при выборе направления подготовки в вузе.

Цель ВКР: разработать информационную систему для автоматизации процесса проведения профориентационного тестирования в вузе.

В первом разделе ВКР описана основная деятельность ХТИ – филиала СФУ и концепция IT-проекта автоматизации бизнес-процесса «Проведение профориентационного тестирования». Представлен анализ методик профориентационного тестирования и профориентационных систем. Обоснован выбор средств разработки информационной системы профориентационного тестирования.

Во втором разделе описан процесс настройки программного окружения необходимый для разработки информационной системы, процесс разработки информационной системы, включая все его этапы.

В третьем разделе была выполнена оценка экономической эффективности программного продукта, произведен анализ потенциальных рисков, возникающих в процессе разработки и эксплуатации, и представлены соответствующие рекомендации по их устранению.

В результате была разработана информационная система профориентационного тестирования.

SUMMARY

The theme of the graduation thesis is «IT System for Career Guidance Testing at KhTI – branch of SibFU». It consists of 90 pages, 10 tables, 46 figures, 8 formulae, 18 references, 2 appendices.

VOCATIONAL TESTING, WEB APPLICATION, IT SYSTEM, RESPOND-ENT, PRE-UNIVERSITY TUTORIAL AND ADMISSION OFFICE, YOUNG EN-GINEER TRAINING CENTER, SPECIALIST OF PRE-UNIVERSITY TUTORIAL AND ADMISSION OFFICE, IDEF3, DFD, PYTHON, FLASK

The object of the thesis is the methods of automating testing, questionnaires, and survey processes.

The subject of the thesis is IT system for vocational testing in choosing a study program at the university.

The purpose of the paper is to develop an IT system to automate the process of conducting vocational testing at the university.

The first section describes the work of KhTI – branch of SibFU and the concept of the IT project for automating the business process of «Conducting Vocational Test-ing». It presents an analysis of vocational testing methodologies and vocational sys-tems. The choice of development tools for the IT system for vocational testing has been determined.

The second section describes the process of setting up the software environment necessary for the development of the IT system and the development process itself, in-cluding all its stages.

The third section includes the calculation of the economic efficiency of the soft-ware product, the analysis of potential risks arising during the development and opera-tion process, and corresponding recommendations for their mitigation.

As a result, the IT system for vocational testing has been developed.

English language supervisor

Signature, date

N.V. Chezybaeva

СОДЕРЖАНИЕ

Введение.....	6
1 Анализ предметной области	9
1.1 Характеристика основной деятельности Хакасского технического института – филиала Сибирского федерального университета.....	9
1.2 Концепция IT-проекта автоматизации бизнес-процесса «Проведение профориентационного тестирования»	14
1.3 Анализ прототипа информационной системы поддержки профориентационного тестирования	18
1.4 Анализ методик профориентационного тестирования прототипа информационной системы	21
1.5 Анализ профориентационных систем, представленных в открытом доступе	23
1.6 Моделирование информационной системы профориентационного тестирования.....	28
1.6.1 Модель «как будет».....	28
1.6.2 Модель взаимодействия пользователя с информационной системой..	31
1.6.3 Модель потока данных	35
1.6.4 Модель «сущность-связь»	37
1.7 Сравнительный анализ и выбор средств разработки веб-приложения «ПрофТест ХТИ»	39
1.7.1 Выбор системы управления базами данных.....	39
1.7.2 Выбор языков программирования	41
Вывод по разделу «Анализ предметной области».....	46
2 Описание процесса разработки информационной системы профориентационного тестирования	47
2.1 Настройка среды разработки.....	47
2.2 Разработка базы данных	49
2.3 Разработка формы регистрации и авторизации пользователей в системе .	51
2.3 Разработка личного кабинета сотрудника ОДПиНН.....	56

2.4 Разработка личного кабинета super-пользователя	64
2.4 Разработка личного кабинета респондента	66
2.5 Разработка и внедрение шаблона писем для респондента.....	70
Вывод по разделу «Описание процесса разработки информационной системы профориентационного тестирования»	73
3 Оценка экономической эффективности разработки и внедрения информационной системы	74
3.1 Капитальные затраты	74
3.1.1 Затраты на проектирование	75
3.1.2 Расчёт капитальных затрат	78
3.2 Расчёт эксплуатационных затрат	80
3.3 Совокупная стоимость владения системой	82
3.4 Оценка рисков реализации проекта.....	83
3.5 Экономическая эффективность реализации проекта	85
Вывод по разделу «Оценка экономической эффективности разработки и внедрения информационной системы».....	87
Заключение	88
Список использованных источников	89
Приложение А	91
Приложение Б.....	98

ВВЕДЕНИЕ

Один из ключевых моментов в жизни человека заключается в выборе своей будущей профессии. Часто этот выбор делается на основе интуиции или на основе временных желаний и увлечений без должного обдумывания. Также не редки ситуации, когда школьники вынуждены принимать решение под давлением необходимости выбора предметов для сдачи Единого государственного экзамена (ЕГЭ). Однако, из-за недостатка понимания в области, в которой они хотят реализоваться, окружение, включая родителей, навязывает им свой выбор. В результате, человек, выбирая профессию, имеет слабое представление о принципах ее работы, не проявляет должного интереса к качественному выполнению своих обязанностей и, как следствие, не может полноценно реализоваться в выбранной сфере.

Возникают ситуации, когда школьники осознанно выбирают профессию самостоятельно. В таких случаях возникает другая проблема, а именно выбор направления подготовки, поскольку в настоящее время их множество, и сложно определить, какой круг профессий охватывает каждая конкретная специальность.

В связи с тем, что ХТИ – филиал СФУ заинтересован в обучении высококвалифицированных кадров, возникла потребность в системе, которая будет помогать определять наибольшую предрасположенность к той или иной профессии с помощью методик профориентационного тестирования, а также помогать с выбором направления подготовки для потенциальных абитуриентов ХТИ – филиала СФУ.

Объект ВКР: методы автоматизации процессов тестирования, анкетирования, опросов.

Предмет ВКР: информационная система профориентационного тестирования при выборе направления подготовки в вузе.

Цель ВКР: разработать информационную систему для автоматизации процесса проведения профориентационного тестирования в вузе.

Для достижения поставленной цели необходимо решить следующие задачи:

1. выполнить анализ деятельности отдела довузовской подготовки и нового набора ХТИ – филиала СФУ (ОДПиНН) в направлении профориентационного тестирования школьников;
2. выполнить анализ существующих методик профориентационного тестирования и выбрать методики для проекта;
3. выполнить анализ аналогичных программных продуктов и обосновать необходимость собственной разработки;
4. выполнить структурное моделирование для описания бизнес-процесса, подлежащего автоматизации;
5. выполнить объектно-ориентированное моделирование с применением унифицированного языка моделирования *UML*;
6. выполнить сравнительный анализ и обосновать выбор программных средств разработки информационной системы;
7. разработать программный продукт;
8. выполнить тестирование ИС;
9. оценить экономическую эффективность разработки и внедрения информационной системы.

В первом разделе ВКР описана основная деятельность ХТИ – филиала СФУ и концепция IT-проекта автоматизации бизнес-процесса «Проведение профориентационного тестирования». Представлен анализ методик профориентационного тестирования и профориентационных систем. Обоснован выбор средств разработки информационной системы профориентационного тестирования.

Во втором разделе описан процесс настройки программного окружения необходимый для разработки информационной системы. Был подробно описан процесс разработки информационной системы, включая все его этапы, а также был описан процесс тестирования окончательного варианта.

В третьем разделе была выполнена оценка экономической эффективности программного продукта, произведен анализ потенциальных рисков, возникающих в процессе разработки и эксплуатации, и представлены соответствующие рекомендации по их устранению.

В процессе выполнения выпускной квалификационной работы были применены различные методы исследования. Для теоретического подхода был выполнен глубокий анализ доступных материалов и литературы, систематизированы полученные данные и сформулированы соответствующие выводы. Кроме того, был использован эмпирический подход, основанный на моделировании, чтобы получить практическое представление о рассматриваемой проблеме и проверить предложенные теоретические концепции и гипотезы на практике. Все эти методы в совокупности позволили успешно провести исследование и достичь поставленных целей выпускной работы.

Система профориентационного тестирования может быть применена в различных ситуациях для обеспечения профессиональной ориентации и научных исследований. Она может использоваться на мероприятиях, организуемых отделом довузовской подготовки и нового набора, чтобы помочь потенциальным абитуриентам выбрать подходящую профессию и образовательный путь. Кроме того, система может быть внедрена в деятельность центра подготовки юного инженера, где она поможет определить таланты и интересы учащихся, чтобы подобрать правильные образовательные планы.

Результаты исследования были представлены на XIX Международной конференции студентов, аспирантов и молодых ученых «Перспективны – 2023», посвященной Международному году фундаментальных наук в интересах устойчивого развития, г. Красноярск, а также на Республиканском конкурсе научно-исследовательских работ студентов «Научный потенциал Хакасии – 2030».

1 Анализ предметной области

1.1 Характеристика основной деятельности Хакасского технического института – филиала Сибирского федерального университета

Создание технического института в Хакасии было обусловлено тем, что с середины 1960-х годов в южной части Красноярского края развивался производственный комплекс Саянского промышленного края. Для успешного решения масштабных задач по строительству в Хакасии требовались значительные изменения в кадровом составе. В связи с этим было принято решение открыть общетехнический факультет (ОТФ) Красноярского политехнического института в городе Абакане. Открытие факультета состоялось 28 февраля 1968 года [1].

Саянский промышленный территориально-производственный комплекс стал стремительно развиваться, что привело к возникновению потребности в новых инженерных специалистах. В результате, в 1972 году общетехнический факультет был преобразован в филиал Красноярского политехнического института в городе Абакане [1].

В 1976 г. была открыта дневная форма обучения по направлениям «Промышленное и гражданское строительство», «Технология машиностроения» и «Электроснабжение промышленных предприятий».

В 1999 году институт претерпел переименование и стал известен как Хакасский технический институт – филиал Красноярского государственного технического университета (ХТИ – филиал КГТУ). В этом же году были добавлены новые специальности, включая «Экспертиза и управление недвижимостью», «Электропривод и автоматика промышленных установок и технологических комплексов» и «Автомобильные дороги и аэродромы». В 2001 году была открыта специальность «Прикладная информатика в экономике». Через восемь лет институт вошел в состав Федерального образовательного учреждения высшего профессионального образования «Сибирский федеральный университет» и получил новое наименование – Хакасский технический институт – филиал

Сибирского федерального университета (ХТИ – филиал СФУ).

Через три года ФГОУ ВПО «Сибирский федеральный университет» был преобразован в автономное учреждение. В результате Хакасский технический институт стал филиалом федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет».

Деятельность ХТИ – филиала СФУ осуществляется на основании законодательства РФ, Федерального закона «Об образовании в Российской Федерации» (№273-ФЗ от 29.12.2012г.), иными постановлениями Правительства РФ, устава СФУ, положения о ХТИ – филиале СФУ и другими локальными актами, а также лицензии на право осуществления образовательной деятельности и свидетельства о государственной аккредитации [1].

Хакасский технический институт – филиал Сибирского федерального университета (ХТИ – филиал СФУ) стремится создавать образовательную, научно-исследовательскую и инновационную инфраструктуру, способствуя развитию социально-экономического потенциала Сибирского федерального округа. Основная цель заключается в продвижении новых знаний и технологий, а также формировании высококвалифицированных кадров, необходимых для решения современных вызовов и задач регионального развития.

В институте проводится обучение в различных формах – очной, очно-заочной и заочной. В рамках специалитета предлагается одно направление, а в бакалавриате – шесть направлений. Также имеется возможность получить образование по одному направлению магистратуры. Институт готовит специалистов и бакалавров в следующих областях: электроэнергетика, машиностроение и материалообработка, строительство, экономика, транспорт и информатика.

В институте также активно развиваются программы дополнительного профессионального образования, включая курсы повышения квалификации, профессиональную переподготовку и тематические семинары для руководителей и специалистов предприятий и организаций [1]. На рисунке 1.1 представлена организационная структура ХТИ – филиала СФУ.

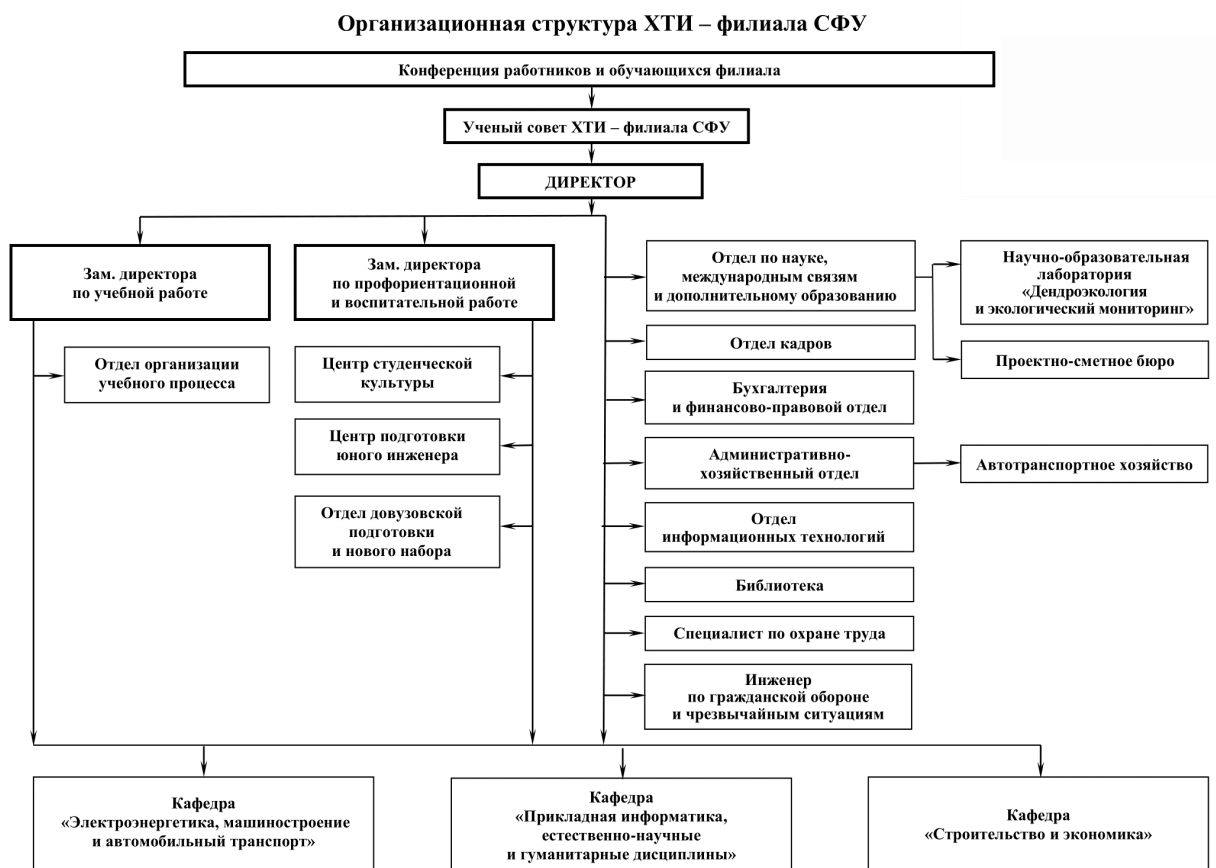


Рисунок 1.1 – Схема структуры ХТИ – филиала СФУ

Помимо официального веб-сайта, сотрудники института и студенческая медиагруппа активно развивают свои группы в социальных сетях *VK* и *Telegram*, а также на *YouTube*-канале, где публикуется информация не только о жизни студентов, но и о важных моментах для абитуриентов. Например, здесь можно найти информацию о предстоящих «Днях Открытых дверей», изменениях в правилах приема на текущий учебный год, сроках подачи заявлений и многое другое.

Ежегодно институт проводит мероприятие «День Открытых дверей». Данное мероприятие проводится с целью привлечения абитуриентов. Абитуриенту предоставляется возможность познакомиться поближе с направлениями подготовки ХТИ – филиала СФУ и их особенностями, задать интересующие его вопросы организаторам данного мероприятия, а также лично побывать в стенах учебного заведения. Побывав на данном мероприятии, абитуриент может по-

нять, хочет ли он обучаться в данном институте, а также имеет возможность определиться с будущим направлением учёбы. Начиная с 2022 года, появилась возможность проведения подобных мероприятий в дистанционном формате, что очень удобно, ведь теперь все абитуриенты, проживающие в удалённых городах, также имеют возможность присутствовать на «Дне Открытых дверей».

Узнав о правилах приёма, абитуриент имеет возможность записаться на обучающие курсы «Центра подготовки юного инженера» (ЦПЮИ), чтобы подготовиться к сдаче «Единого государственного экзамена» (ЕГЭ) и поступить в институт.

ЦПЮИ является самостоятельным структурным подразделением ХТИ – филиала СФУ. Центр создан в соответствии с Уставом ФГАОУ ВПО «Сибирский федеральный университет» и Положением о ХТИ – филиале СФУ на неограниченный срок деятельности. Местоположение ЦПЮИ: г. Абакан, ул. Щетинкина, 27, ауд. 203 [2].

Структура Центра подготовки юного инженера может быть подвержена изменениям в соответствии со штатным расписанием, утвержденным директором института, и особенностями деятельности самого института.

В структуру ЦПЮИ входят: заведующий центра, который осуществляет руководство работой ЦПЮИ и непосредственно подчиняется директору, и специалист [2].

Центр подготовки юного инженера фокусируется на разработке и распространении дополнительных образовательных курсов, ориентированных на следующие цели:

- повышение уровня знаний учащихся 8 – 11 классов средних образовательных школ по математике, физике, информатике, обществознанию, русскому языку и физике;
- подготовке к олимпиадам по математике и физике;
- организации внеурочной занятости школьников;
- помощь учащемуся в профессиональной ориентации;
- подготовку к сдаче экзамена ОГЭ и ЕГЭ по физике, математике, рус-

скому языку, информатике и обществознанию, а также внутренних вступительных испытаний в вузе.

В рамках учебных программ также предусмотрен лабораторный практикум, который позволяет учащимся активно работать в лабораториях института [2]. Через выполнение лабораторных работ, учащиеся развивают навыки проведения экспериментов и получают практический опыт работы с физическими приборами. Работая вместе, школьники осваивают навыки коллективной работы и научаются самостоятельно делать выводы, основываясь на экспериментальных данных, что способствует более глубокому усвоению теоретического материала.

Также стоит более подробно ознакомиться с деятельностью отдела довузовской подготовки и нового набора (ОДПиНН), так как данный отдел напрямую взаимодействует с абитуриентами.

Руководство деятельностью ОДПиНН осуществляет начальник отдела довузовской подготовки и нового набора. В структуру входят начальник и специалисты. Также данному отделу могут быть подчинены структуры подразделения и временные коллективы, комиссии, образованные для решения отдельных задач, в том числе приёмная комиссия работающая на площадке ХТИ – филиала СФУ [3].

Для успешного формирования контрагентов студентов, а также повышения престижа обучения в ХТИ – филиале СФУ, ОДПиНН разрабатывает стратегию и тактику для подготовки к приёму, проведению профориентационной работы, а также работы по довузовской подготовке и вступительных испытаний.

К основным задачам отдела довузовской подготовки можно отнести следующие:

- организация проведения рекламной кампании по набору на первый курс, а также разработка и утверждение рекламно-информационных материалов;
- формирование плана общеинститутской работы по новому набору сту-

дентов;

- организация работы экзаменационной, апелляционной комиссии на площадке ХТИ – филиала СФУ;

- организация и контроль работы технического секретариата во время приёмной кампании;

- осуществление работы с автоматизированной информационной системы (АИС) «Абитуриент СФУ» [3].

Отдел ОДПиНН активно организует более ста мероприятий в год для работы с абитуриентами, что способствует эффективной реализации поставленных задач. Однако, стоит отметить, что с осени 2019 года профориентационное тестирование среди абитуриентов (далее респондентов) не проводится.

Респонденты могут использовать профориентационное тестирование для более осознанного выбора области обучения в вузе, а проведение этого тестирования поможет специалистам провести консультацию с респондентами. Анализ основных функций ОДПиНН и консультации с его специалистами привели к необходимости разработки автоматизированной системы профессионального тестирования.

1.2 Концепция IT-проекта автоматизации бизнес-процесса «Проведение профориентационного тестирования»

Выбор будущей профессии является одним из важных решений в жизни каждого человека. Ежегодно учащиеся старших классов сталкиваются с проблемой выбора профессии из-за большого разнообразия на рынке труда. Им сложно определить сферу деятельности, к которой они наиболее предрасположены, как следствие, возникают сложности при выборе предметов для единого государственного экзамена, что приводит к увеличению стресса у ученика. Одним из вариантов решения данной проблемы, является прохождение профориентационного тестирования, которое поможет выявить интерес к определённой сфере деятельности, а также определит перечень профессий, который суще-

ствуется в ней.

В последние годы профориентационная деятельность стала одним из приоритетов развития образования на региональном и государственном уровнях. Обучение профессионалов является одним из важнейших рычагов экономического роста страны, поэтому власти активно разрабатывают и реализуют меры, направленные на создание основы для первоначальной профессиональной ориентации [4].

Профориентация представляет собой комплекс действий для определения склонностей и талантов в определенных видах профессиональной деятельности у человека, а также система действий, признанных помочь выбрать карьерный путь [5]. Инструменты профессионального руководства должны способствовать самопознанию учащихся и помогать им серьезно относиться к своему будущему, исследуя различные варианты профессий.

В российских школах уже существует система профориентации (например, экскурсии на предприятия, с помощью которых старшеклассники знакомятся с разными профессиями).

Эксперты отмечают, что до сих пор осведомленность о перспективах карьерного роста среди учащихся в школе очень низкая. Несмотря на то, что каждый второй ученик средней школы проходит профтестирование, чтобы определить его профессиональные интересы, каждый пятый ученик утверждает, что школа никоим образом не помогла ему с выбором его будущей профессии. В то же время школьники отмечают, что большинство уроков профориентации скучны. В лучшем случае они ограничиваются несколькими короткими поездками на городские предприятия, которые носят вполне формальный характер [5].

Статистика показывает, что тридцать процентов учащихся имеют лишь смутные представления о возможностях трудоустройства, а также неправильно оценивают свои навыки, интересы относительно подходящей профессии [6]. Многие люди поступают в высшие учебные заведения, независимо от специальности, к которой они наиболее подготовлены или к которой имеют есте-

ственную склонность. В результате неправильный выбор университета и направления обучения часто приводит к потраченным впустую годам. Другими словами, молодые люди обучаются в предметной области, где они, очевидно, никогда не будут работать.

Школьникам важно помочь определить приоритетную для них сферу будущей деятельности, ведь многие из них по окончании школы не могут определиться кем они хотят быть и в какой области развиваться в дальнейшем.

Зачастую выбор делается на основании индивидуальных предпочтений или на основании потребностей регионального рынка труда, поэтому сложно эффективно решать данную задачу без применения автоматизированной системы. Данная система позволит проводить тестирование среди большого количества учащихся, таким образом, автоматизирует процесс проверки результатов тестирования, а также поможет точнее определить индивидуальные особенности и интересы выпускников школы и поможет сделать правильный выбор будущей профессии.

Так как профориентационное тестирование зачастую проводится одновременно среди большого количества учащихся, для эффективной и быстрой обработки результатов тестирования необходимо использовать информационные технологии.

ХТИ – филиал СФУ в настоящее время не имеет в своём распоряжении рабочей автоматизированной системы профтестирования, которая бы позволила провести профориентационное тестирование среди респондентов с конкретным предложением выбора направления.

В настоящее время в отделе довузовской подготовки и нового набора ХТИ – филиала СФУ не используется автоматизированная система профориентационного тестирования, но она необходима для проведения профориентационной кампании, привлечения респондентов и т.д. В связи с чем, возникла идея разработки данной системы тестирования, которая будет реализована в виде веб-приложения.

Планируется использование данной системы на таких мероприятиях как:

«День открытых дверей» и на выездных мероприятиях для привлечения респондентов.

Исходя из вышесказанного, создание системы для автоматизированного профориентационного тестирования на сегодняшний день является актуальным. Веб-приложение для профессионального тестирования «ПрофТест ХТИ» предназначено для решения следующих задач:

- проведение онлайн тестирования респондентов;
- предоставление результатов профориентационного тестирования;
- ведение базы данных участников тестирования для рассылки информации о курсах, предоставляемых «Центром подготовки юного инженера».

Система должна обладать следующим функционалом:

- возможность добавления и открытия доступа к готовым электронным тестам по профориентации, составленные на основе общепринятых методик;
- возможность отправки результатов тестирования участника на его электронную почту, которая введена при входе в систему;
- возможность выгрузки отчётов тестирования специалистом;
- возможность проведения тестирования в любое время и в любом месте (школа или институт).

Веб-приложение позволяет работать в многопользовательском режиме, таким образом, интерфейс и функционал системы будут адаптированы под задачи пользователя. Система будет включать в себя трёх пользователей: *super*-пользователь, специалист ОДПиНН и респондент. Подробнее об их взаимодействии с системой описано в пункте 1.7.

Использование веб-приложения в профориентационной деятельности предоставляет следующие преимущества:

- повышение качества психологической поддержки в образовательном процессе благодаря использованию современных и проверенных методик;
- быстрая обработка и визуализация данных, что сокращает время интерпретации результатов теста;

– снижение вероятности ошибок в результате расчета психодиагностического исследования;

– автоматизация процесса составления психологического диагноза.

Таким образом, разработка и внедрение системы «ПрофТест ХТИ» позволит повысить уровень качества и эффективность профориентационной работы сотрудников ОДПиНН.

1.3 Анализ прототипа информационной системы поддержки профориентационного тестирования

Ранее для процесса профессиональной ориентации в институте использовалась программа, которая была разработана студентом направления 09.03.03 «Прикладная информатика» Бредюк П. А. Данная ИС будет служить прототипом для разработки веб-приложения «ПрофТест ХТИ».

Интерфейс и логика приложения были разработаны на языке программирования *Delphi*. Данную ИС можно использовать на локальном компьютере после установки, что является не совсем удобным решением на сегодняшний день, так как не всегда имеется возможность осуществить установку ИС на компьютеры, где планируется проведение мероприятия для будущих абитуриентов.

Интерфейс приложения является устаревшим и не совсем удобным для взаимодействия с пользователем (рис 1.2). В верхней части окна расположены вкладки методик тестирования. Пользователь не имеет возможности самостоятельно переключаться между вкладками. Переход осуществляется автоматически после того, как пользователь пройдет тестирование и получит результат по методике, которая представлена на текущий момент на экране пользователя. Также стоит обратить внимание на блок интерфейса, где представлены варианты ответов. Кнопки, представленные в интерфейсе, являются не совсем удобными из-за своего размера, так как пользователю необходимо прицеливаться

курсором мыши, чтобы выбрать необходимый ответ. Всё это доставляет некоторое неудобство при взаимодействии с данной информационной системой.

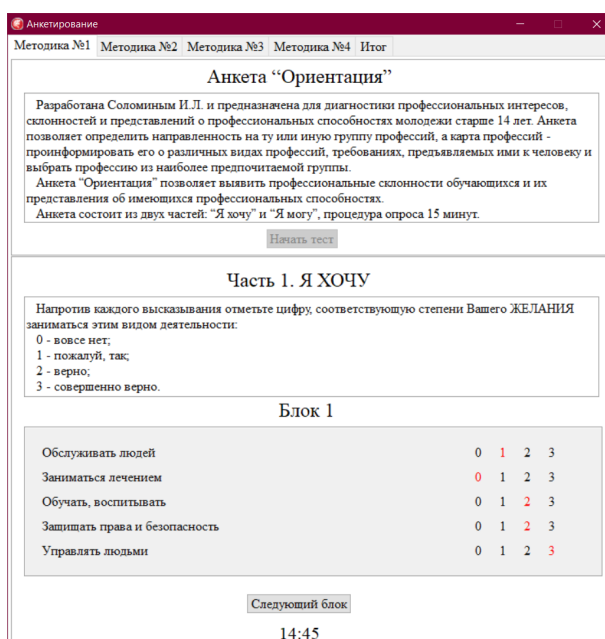


Рисунок 1.2 – Интерфейс прототипа ИС

Результат тестирования отображается в отдельном окне. В зависимости от методики представлен либо в виде перечня профессий (рис. 1.3), либо текста, который подробно описывает тип личности респондента, а также приводит небольшой перечень подходящих профессий (рис. 1.4).

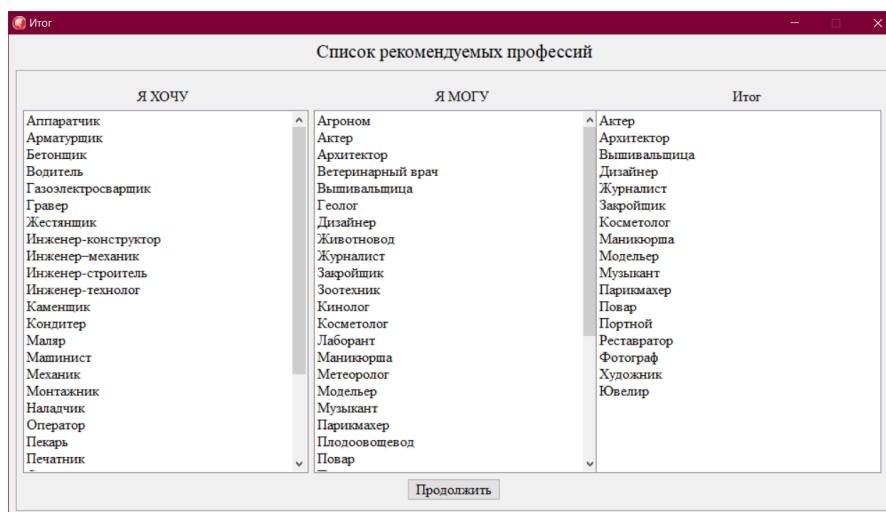


Рисунок 1.3 – Результат тестирования по анкете «Ориентация»

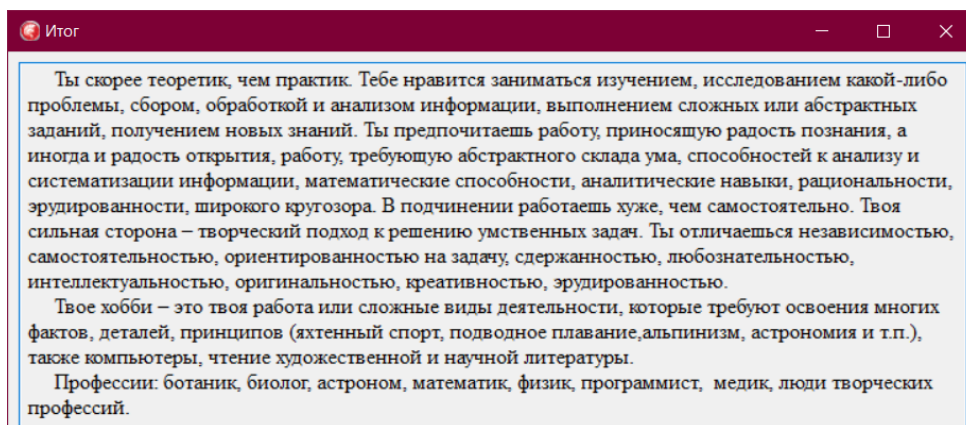


Рисунок 1.4 – Результат тестирования по опроснику «Профессиональных предпочтений»

Также в системе формируется итоговый результат на основе результатов тестирования по четырём методикам. В итоговом отчёте пользователь видит перечень приоритетных сфер деятельности, а также перечень рекомендуемых профессий (рис. 1.5).

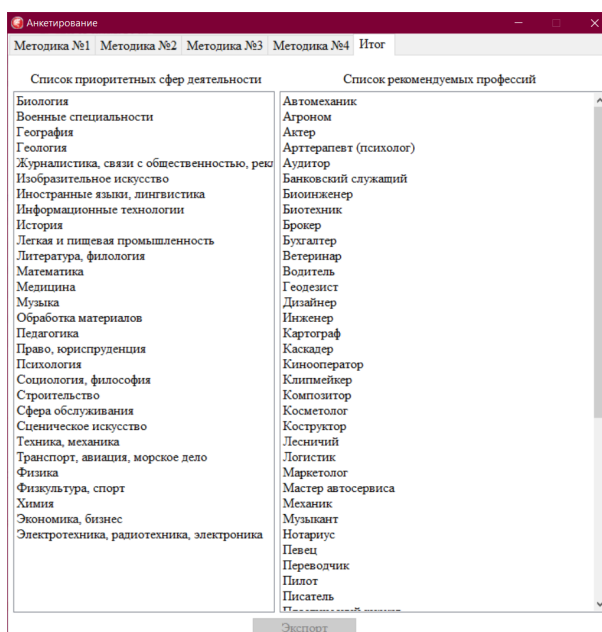


Рисунок 1.5 – Итоговый результат тестирования

Таким образом, прототип не удовлетворяет современным тенденциям развития информационных систем. Информационная система имеет следующие недостатки:

- не актуальный дизайн;
- отсутствие удалённого доступа к ИС;
- отсутствие авторизации пользователя;
- отсутствие наличия отчётов по результатам тестирования для сотрудника ОДПиНН;
- отсутствие самостоятельного выбора методики для тестирования.

Данные недостатки, имеющиеся в данной системе, планируется исключить при разработке веб-приложения для профориентационного тестирования.

1.4 Анализ методик профориентационного тестирования прототипа информационной системы

На сегодняшний день психологами разработано множество профориентационных методик. Для проведения профориентационного тестирования респондентов, необходимо выбрать методики, которые имеют чёткую профессиональную направленность, а также имеют научное обоснование. Наиболее известными и признанными среди российских психологов являются следующие методики:

- Анкета «Ориентация» (разработана Соломиным И.Л.).
- Опросник профессиональных предпочтений (типология Дж. Холланда).
- Методика «Карта интересов» (разработана Голомшток А. Е.).
- «Матрица выбора профессии» (методика Резапкиной Г.).

Анкету «Ориентация» разработал петербургский психолог Соломин Игорь Леонидович. Данную методику применяют для самооценки профессиональных интересов и способностей респондентов. Психологи рекомендуют пройти тестирование с помощью анкеты тем людям, которые затрудняются с выбором профессии. Данный вопрос актуален как для выпускников школ, так и для взрослых людей, которые находятся в поиске работы. В ходе тестирования респондент должен ответить на 70 вопросов, выбрав один из четырех возмож-

ных ответов. По завершении тестирования респондент получает краткую характеристику о своём типе личности [7].

Опросник профессиональных предпочтений был разработан американским учёным и профессором психологии Дж. Холландом, чтобы исследовать профессиональные предпочтения человека. С помощью теста респондент может определить свои склонности и свой интерес к определённым профессиям, а также сузить круг запросов в профессиональной сфере, что поможет планированию профессиональной карьеры. Результатом тестирования является определение типа личности респондента, который характеризует профессиональные навыки, на основе которых стоит выбирать будущую профессию [8].

Методика «Карта интересов», разработанная советским психологом Голомштоком А.Е., была модифицирована психологами А.А. Азабель, А.Г. Грецова. Данная методика в свою очередь помогает узнать респонденту, какие интересы у него являются наиболее выраженными, а также помогает с выбором будущей профессии. В методику включены 144 вопроса, которые разделены на 24 категорий. С помощью данных категорий определяется склонность личности к той или иной профессии [9].

Методика «Матрица выбора профессии» была разработана психологом Галиной Владимировной Резапкиной. Методика содержит в себе вопросы, которые разбиты на две категории: «сфера труда» и «вид труда», каждая из которых содержит в себе 10 вариантов ответов. Респондент должен в каждой категории выбрать от 1 до 3 ответов. На основании выбранных ответов респондент получает список рекомендуемых профессий [10].

Таким образом, было принято решение использовать для профориентационного тестирования методики, описанные выше, так как они позволяют выявить основные виды профессиональной деятельности, которые будут соответствовать направленности и личностным особенностям респондента.

При проектировании проекта планируется улучшить визуальное оформление тестов, в частности вывод результатов тестирования.

1.5 Анализ профориентационных систем, представленных в открытом доступе

Данная система на сегодняшний день имеет несколько аналогичных проектных решений. На рынке программных продуктов в достаточном количестве представлены системы профориентационного тестирования. В связи с чем рассмотрим основной функционал систем и возможность использования в ХТИ – филиале СФУ для респондентов в «День Открытых дверей». Некоторые решения носят коммерческий характер, а некоторые из них бесплатны. Анализ аналогов перед началом разработки необходим, для того чтобы определить типичный функционал подобных систем, а также посмотреть как на положительные, так и отрицательные стороны готовых решений.

В данном разделе описан анализ таких готовых решений, как:

- сервис «Тестометрика» [11];
- сервис «Работа России» [12]

Сервис «Тестометрика» является платформой для размещения тестов, разработанных по различным методикам.

Данный сервис содержит в себе огромное количество тестов различной направленности: профессионально психологические, образовательные, а также развлекательные. В основном все предложенные тесты имеют открытый доступ и являются бесплатными для всех пользователей, однако существуют и платные варианты. Например, прежде чем пройти «Расширенный тест на профориентацию» пользователю необходимо будет зарегистрироваться и оплатить стоимость теста, которая указана в описании.

Одним из основных плюсов данного сервиса является лаконичный и современный дизайн (рис. 1.6).

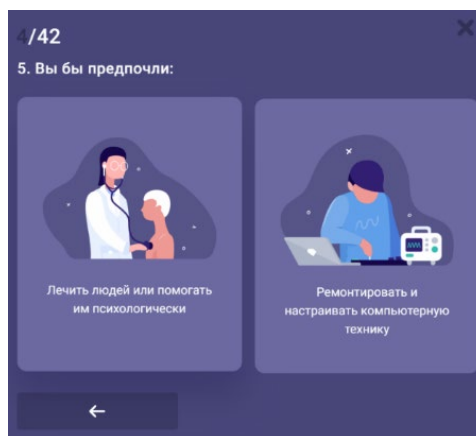


Рисунок 1.6 – Интерфейс теста на определение профориентации сервиса «Тестометрика»

На сервисе представлено большое количество визуального контента, которое привлекает внимание пользователя и помогает ему проще взаимодействовать с сервисом. Также стоит отметить интерактивность данной системы, которая визуально демонстрирует реакцию на те или иные действия пользователя, таким образом, оживляя её.

Так как сервис содержит в себе огромное количество тестов, для быстрого поиска нужного, все тесты разбиты на категории. Чтобы выбрать категорию, пользователю необходимо нажать на одноименную вкладку и выбрать нужную категорию тестов. Также пользователь может отфильтровать список тестов по следующим критериям: по времени прохождения, по статусу выполнения, а также просмотреть весь список тестов.

При желании пользователь может также отсортировать тесты: по популярности, по новизне и по алфавиту.

Ещё одним преимуществом системы является то, что при прохождении теста система запоминает на каком вопросе был остановлен процесс тестирования. При закрытии теста, ему присваивается статус «В процессе», а также он отмечается специальным значком (рис. 1.7). Таким образом, открыв тест, пользователю не придётся заново проходить тест с самого начала, а можно будет продолжить отвечать на вопросы с того места, на котором была закончена предыдущая сессия.

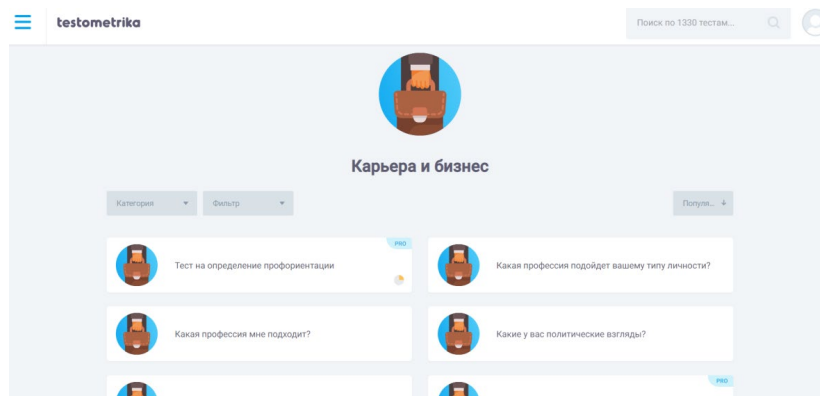


Рисунок 1.7 – Список тестов категории «Карьера и бизнес»

К минусам системы можно отнести то, что все тесты одной группы обозначаются одним графическим обозначением (рис. 1.7). Это не совсем удобно, так как при просмотре списка тестов пользователю необходимо постоянно концентрировать своё внимание на названии теста, что довольно сложно, ведь изначально взгляд цепляется именно за картинку.

Ещё одним минусом является то, что все бесплатные профориентационные тесты имеют обобщенный характер. Пройдя их, пользователь сможет лишь приблизительно узнать какая профессия подходит ему, однако это не поможет ему в выборе специальности при поступлении в ВУЗ, так как система не предоставляет список специальностей, которые помогают освоить ту или иную профессию, а также список ВУЗов, которые обучают по данным направлениям.

Сервис «Тестометрика» является хорошим решением для размещения и прохождения тестирования по различным направлениям. Основным недостатком данного сервиса является представление обобщенных результатов тестирования, направленных на профориентацию.

Следующий сервис – «Работа России». Данный сервис является федеральной государственной системой Федеральной службы по труду и занятости. Данная система позволяет составить резюме, а также помогает осуществлять поиск определенных вакансий. Таким образом, выполняет свою основную функцию – помощь в трудоустройстве.

Однако несмотря на то, что «Работа России» является площадкой для

размещения и поиска вакансий, она также предоставляет пользователям возможность пройти профориентационное тестирование (рис. 1.8) по следующим направлениям:

- выбор профессии;
- индивидуально-психологические особенности;
- ценностные ориентации в карьере;
- предпринимательская деятельность.

При прохождении тестирования были выявлены следующие преимущества системы:

– пользователь может прикрепить результаты тестирования в своё резюме, таким образом, сохранить их;

– например, при прохождении теста «Выбор профессии» пользователь не только узнаёт к какому типу человека у него наибольшая предрасположенность, но и получает список профессий, которые являются наиболее подходящими для него.

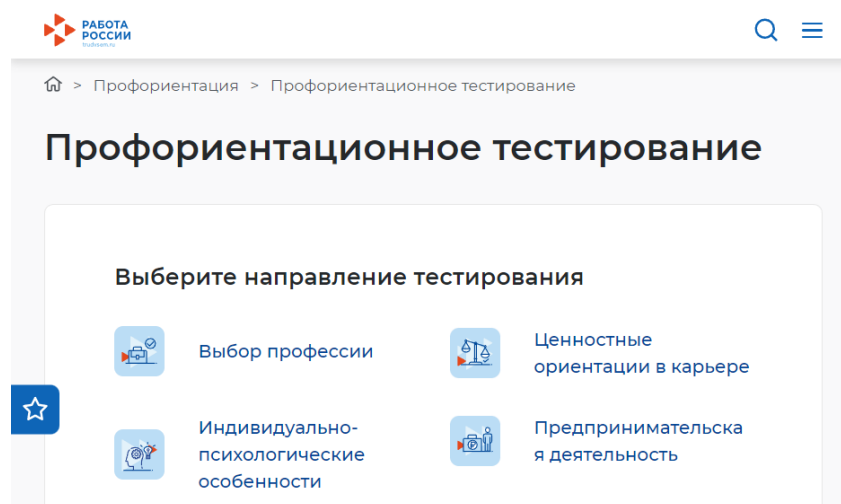


Рисунок 1.8 – Раздел профориентации на сервисе «Работа России»

Недостатками системы являются:

– Дизайн окна для тестирования (рис. 1.9). Элемент, отвечающий за визуализацию количества пройденных вопросов, имеет не самое удачное располо-

жение, а также слишком большой размер, из-за этого вопрос, на который необходимо ответить пользователю, визуально теряется на общем фоне. Пользователю каждый раз необходимо концентрировать свой взгляд, чтобы найти и прочитать вопрос.

The image shows a mobile application interface for a "Дифференциально-диагностический опросник" (Differential Diagnostic Questionnaire). At the top, the title is displayed with a close button (X) on the right. Below the title, there is a section labeled "Описание теста" (Test Description) with a blue arrow icon on the left. A progress bar indicates the current position, showing "6 из 20" (6 of 20) items. Underneath, the text "Я предпочитаю:" (I prefer:) is followed by two selectable options in rounded rectangular boxes. The first option is "Копировать рисунки, изображения (или настраивать музыкальные инструменты)." (Copy drawings, images (or adjust musical instruments)). The second option is "Управлять каким-либо грузовым (подъемным или транспортным) средством (подъемным краном, трактором, телевизоров, и др)." (Control any kind of cargo (lifting or transport) vehicle (lifting crane, tractor, televisions, etc.)).

Рисунок 1.9 – Дизайн формы для тестирования

– Тесты направлены на категорию граждан, у которых уже есть образование и которые находятся в поиске работы. Школьникам данные тесты, возможно помогут понять какие для себя стоит рассматривать профессии, однако это не поможет им в выборе направления обучения, так как данная информация отсутствует, а пользователь не всегда знает, какое высшее образование ему необходимо получить, чтобы в дальнейшем работать по определённой профессии.

Сервис «Работа России» отлично подойдёт людям, которые находятся в поиске работы, однако выпускникам школы, которые ещё не определились с тем, кем они хотят быть, тестирование на данном сервисе не поможет понять какое высшее образование им нужно получить, чтобы начать работать. Это является основной проблемой для каждого второго школьника, так как, выбирая ту или иную специальность, он не понимает, где и на какой должности он может работать в будущем.

Рассмотренные аналоги «Тестометрика» и «Работа России» являются не-

плохими решениями, которые частично решают поставленную задачу, но при этом имеют некоторые недостатки, которые не позволяют выбрать их в качестве системы для профориентационного тестирования ХТИ – филиала СФУ. Кроме того, данные системы не предоставляют необходимые настройки ограничений доступа к создаваемым материалам. Таким образом, было принято решение разработать собственную ИС.

1.6 Моделирование информационной системы профориентационного тестирования

Основная цель моделирования информационной системы заключается в создании четкого представления о разрабатываемой системе и процессах, которые она автоматизирует. Моделирование помогает осуществить проектирование и анализ внутренних процессов системы, а также рассмотреть движение потока данных внутри системы и пользовательские сценарии ее использования.

1.6.1 Модель «как будет»

Так как процесс тестирования не будет подлежать изменению, было принято решение рассмотреть его сразу на модели бизнес-процесса, которая покажет взаимодействия пользователей с системой в рамках дня открытых дверей.

В системе будет предусмотрено три типа пользователя: *Super*-пользователь – данный пользователь будет открывать доступ к методикам тестирования для респондентов; специалист ОДПиНН – сотрудник, который будет вести статистику результатов тестов внутри системы; респондент – школьник, который будет проходить профориентационное тестирование. Для наглядного представления бизнес-процесса с точки зрения отдельного пользователя, создана модель *Role Activity Diagram* (рис. 1.10).

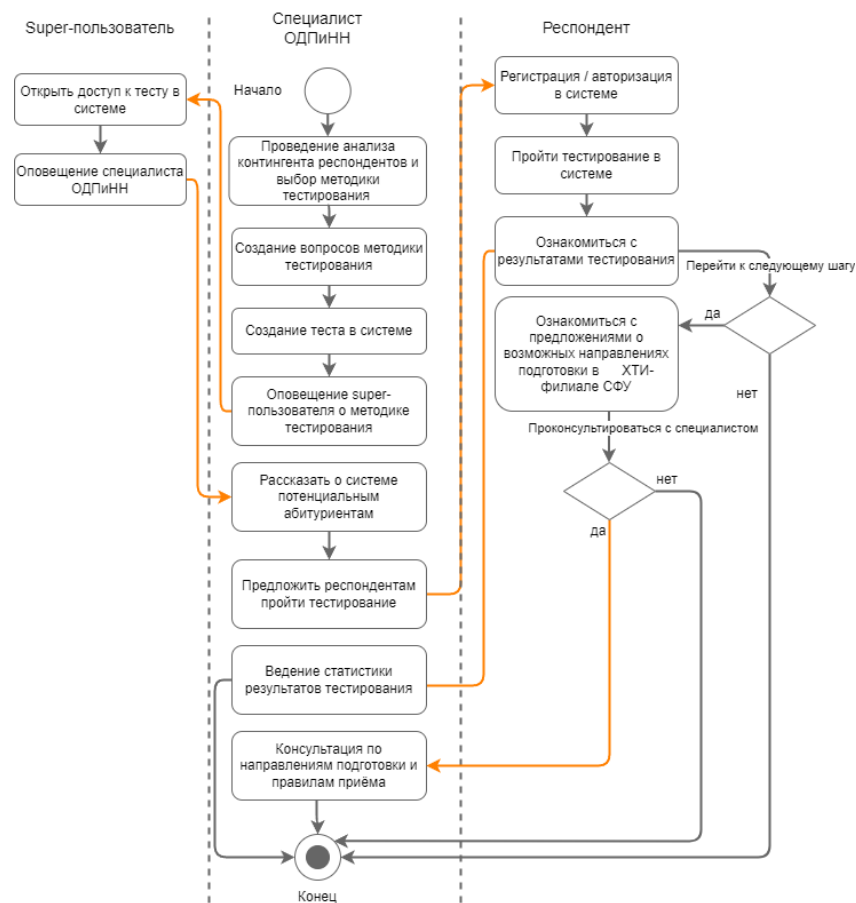


Рисунок 1.10 – Модель бизнес-процесса организации профориентационного тестирования «Как будет»

Данная модель наглядно отображает процесс с точки зрения отдельных ролей, групп ролей и взаимодействия ролей в процессе. Под ролью принято понимать абстрактный элемент процесса, который выполняет определенную организационную функцию. Таким образом, с помощью данной модели можно визуальнo отобразить степень «ответственности» за процесс и его операции, а также взаимодействия ролей [13].

Процесс «Проведение анализа контингента респондентов и выбор методики тестирования» подразумевает, что специалист ОДПиНН за несколько дней до дня открытых дверей должен определить, по какой методике будет проводиться тестирование респондентов. Для этого он проводит анализ потенциальных респондентов по некоторым критериям, смотрит на общий план проведения мероприятия и, исходя из полученных результатов, выбирает наиболее

оптимальный вариант тестирования по определённой методике. Процесс «Создание вопросов методики тестирования» подразумевает, что сотрудник открывает в своём личном кабинете страницу для создания вопросов. С помощью формы, разработанной для создания вопросов внутри системы, пользователь вносит данные вопросов методики тестирования в базу данных (БД). После внесения данных в БД о вопросах методики тестирования, сотрудник ОДПиНН переходит к процессу «Создание теста в системе». Данный процесс подразумевает, что сотрудник открывает страницу, где представлены все созданные им вопросы и на их основе он создаёт новый тест в системе. Следующий процесс, который выполняет специалист: «Оповещение *super*-пользователя о методике тестирования», т. е. он составляет служебную записку, в которой указывает, когда, где и по какой методике будет проводиться тестирование, чтобы *super*-пользователь в свою очередь успел подготовить систему к работе.

Получив от специалиста ОДПиНН служебную записку и ознакомившись с ней, *super*-пользователь начинает выполнять процесс «Открыть доступ к тесту в системе». В начале он разрабатывает алгоритм вычисления результата для методики тестирования, данные для которой ввёл сотрудник ОДПиНН, а после публикует тест в системе, чтобы респондент мог пройти тестирование. При проведении мероприятий, организованных ОДПиНН, *super*-пользователь по запросу сотрудника ОДПиНН скрывает некоторые методики тестирования и оставляет доступ только для тех, что нужны для проведения профориентационного мероприятия. Выполнив данный процесс, *super*-пользователь переходит к следующему процессу: «Оповещение специалиста ОДПиНН о готовности работы», чтобы специалист мог дальше работать с системой.

В обязанности специалиста ОДПиНН в «День Открытых дверей» входит «рассказать о системе потенциальным респондентам», чтобы они могли узнать, для чего нужна данная система и какие у неё есть преимущества. Выполнив данный процесс, следующим шагом специалиста будет предложить учащимся пройти тестирование в системе. Для этого школьники должны будут перейти в аудиторию, оснащённую необходимым техническим оборудованием. В обязан-

ности специалиста входит подготовить заранее компьютеры для работы в системе и провести краткий инструктаж для школьников перед началом работы. После проведения тестирования, специалист обязан выполнить анализ результатов тестирования, чтобы определить какой процент из респондентов, прошедших тестирование, могут быть потенциальными респондентами ХТИ – филиала СФУ в текущем году.

Респонденту необходимо перед началом тестирования зарегистрироваться в системе. При регистрации пользователь указывает адрес электронной почты, на который в дальнейшем можно будет отправить сообщение с результатами тестирования, а также получать уведомления от Центра подготовки юного инженера с информацией о курсах. Далее респондент авторизуется и проходит представленный в системе тест. По окончании тестирования респондент может ознакомиться с результатами теста. Кроме того, после окончания тестирования, в случае если учащийся заинтересован в возможности поступления в ХТИ – филиал СФУ, ему будет предоставлена возможность ознакомиться с актуальной информацией о направлениях подготовки данного института. Респонденту будут представлены контактные данные специалиста, для дальнейшей консультации по вопросам правил приёма на текущих год.

Таким образом, работа в системе была рассмотрена с точки зрения отдельного пользователя, что помогло рассмотреть все внутренние процессы системы, а также разобраться в взаимодействии между ними.

1.6.2 Модель взаимодействия пользователя с информационной системой

На основе рассмотренных выше внутренних процессов в системе, для каждого пользователя была построена модель бизнес-процесса *IDEF3*, которая помогает наглядно продемонстрировать пользовательский сценарий отдельного пользователя при взаимодействии с системой.

Первой была создана модель специалиста ОДПиНН (рис. 1.11), так как он

ответственен за первые этапы создания профориентационного теста в системе.

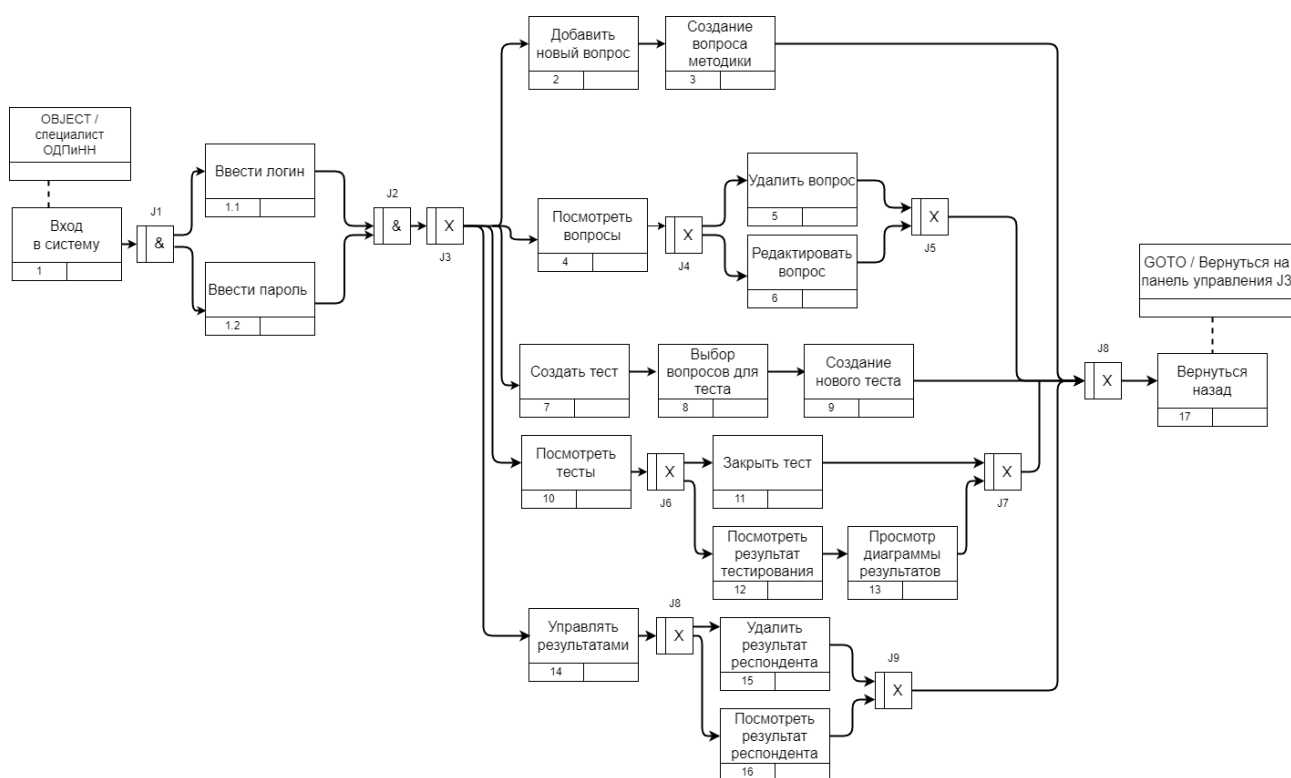


Рисунок 1.11 – Диаграмма работы в системе специалиста ОДПиНН

Специалисту ОДПиНН необходимо авторизоваться в системе, чтобы начать в ней работать. После успешного входа в систему, пользователь попадает на панель управления, где представлены ссылки в виде кнопок на страницы, которые необходимы для выполнения всех функций пользователя:

- добавить вопрос;
- посмотреть вопросы;
- создать тест;
- посмотреть тесты;
- управлять результатами.

Первым делом специалисту необходимо создать в системе вопросы методики тестирования. Для этого ему необходимо нажать на кнопку «Добавить вопрос» и перейти на страницу с формой. После этого специалисту необходимо открыть страницу «Создать тест» и выбрать из перечня вопросов нужные для создания теста по соответствующей методике тестирования. Далее за процесс

создания отвечает *super*-пользователь.

Кроме того, сотрудник имеет возможность просматривать перечень созданных вопросов, удалять, а также редактировать их при необходимости.

Ещё одной основной функцией сотрудника ОДПиНН является просмотр результатов тестирования, как для всех пользователей, так и для каждого респондента, в частности.

Для просмотра обобщённой статистики результата по тесту сотрудник должен нажать кнопку «*Посмотреть тесты*», чтобы перейти на страницу, где будет представлен перечень существующих в системе тестов. Далее он находит нужную методику и нажимает кнопку «*Посмотреть результаты*», чтобы перейти на страницу, где будет построена диаграмма на основе полученных результатов пользователей, которые прошли тестирование по данной методике. Чтобы посмотреть результат определённого респондента, сотрудник должен нажать на кнопку «*Управлять результатами*». На данной странице представлен перечень респондентов, которые зарегистрированы в системе. Для каждого из них указаны методики профориентационного тестирования, по которым было выполнено тестирование данного пользователя. Сотрудник ОДПиНН может посмотреть результат пользователя по методике тестирования либо в случае необходимости удалить его.

Стоит также отметить, что специалист имеет возможность выйти из веб-приложения на любом этапе своей работы.

Следующей была построена модель для *Super*-пользователя (рис. 1.12). Прежде чем начать работать в системе, *Super*-пользователю необходимо войти в систему, ведь без авторизации у него не будет достаточно прав для выполнения своей работы. После входа данный пользователь попадает на страницу, где расположен список тестов, которые необходимо обработать. Далее пользователь выбирает методику тестирования, к которой необходимо предоставить доступ в соответствии с указаниями специалиста ОДПиНН, и кликает по соответствующей кнопке. В случае, если тест не был ранее создан в системе, пользователю необходимо разработать для него алгоритм вычисления результата в соот-

ветствии с методическими рекомендациями, а после опубликовать тест в системе. Выполнив работу, пользователь выходит из системы.

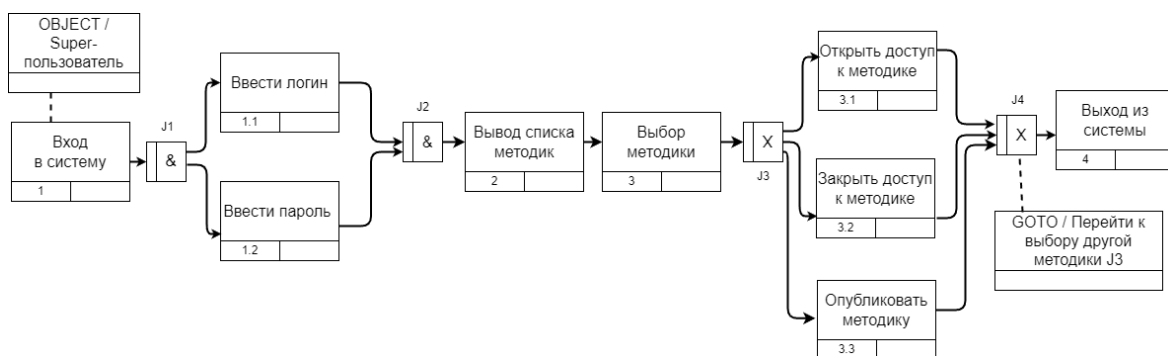


Рисунок 1.12 – Диаграмма работы *Super*-пользователя в системе

Последней была создана модель работы респондента (рис. 1.13). Прежде чем пройти профориентационное тестирование пользователь должен зарегистрироваться в системе, указав при регистрации адрес электронной почты, на которую можно будет сохранить результат, а в дальнейшем получать уведомления о курсах подготовки ЦПЮИ. Авторизовавшись в системе, пользователь открывает методику профориентационного тестирования, чтобы ответить на вопросы. По завершении тестирования, респондент может ознакомиться с полученным результатом. При желании пользователь может пройти тестирование повторно.



Рисунок 1.13 – Диаграмма работы в системе респондента

Таким образом, рассмотрено три пользовательских сценария: *Super*-пользователь, специалист ОДПиНН и респондент. Для каждого из пользовате-

лей была построена цепочка действий, а также описаны особенности выполнения тех или иных шагов.

1.6.3 Модель потока данных

Для наглядного представления, как выглядит поток данных внутри системы при прохождении профориентационного тестирования респондентом была построена диаграмма *DFD*, которая представлена на рисунке 1.14. Данная диаграмма построена на основе одной сущность – «Респондент» и одного процесса – «Прохождение тестирования».



Рисунок 1.14 – Диаграмма потока данных (*DFD*) респондента

Чтобы рассмотреть процесс «Прохождение тестирования», была выполнена декомпозиция данного процесса (рис. 1.15).

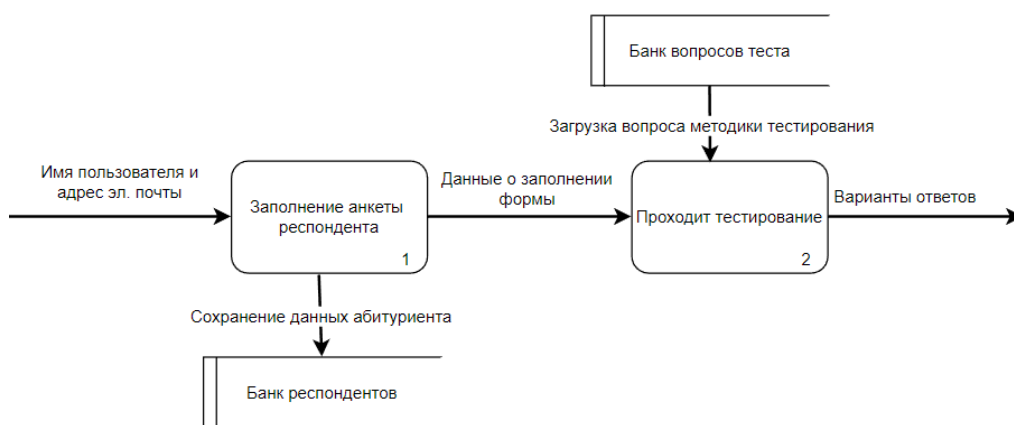


Рисунок 1.15 – Декомпозиция блока 1, лист 1

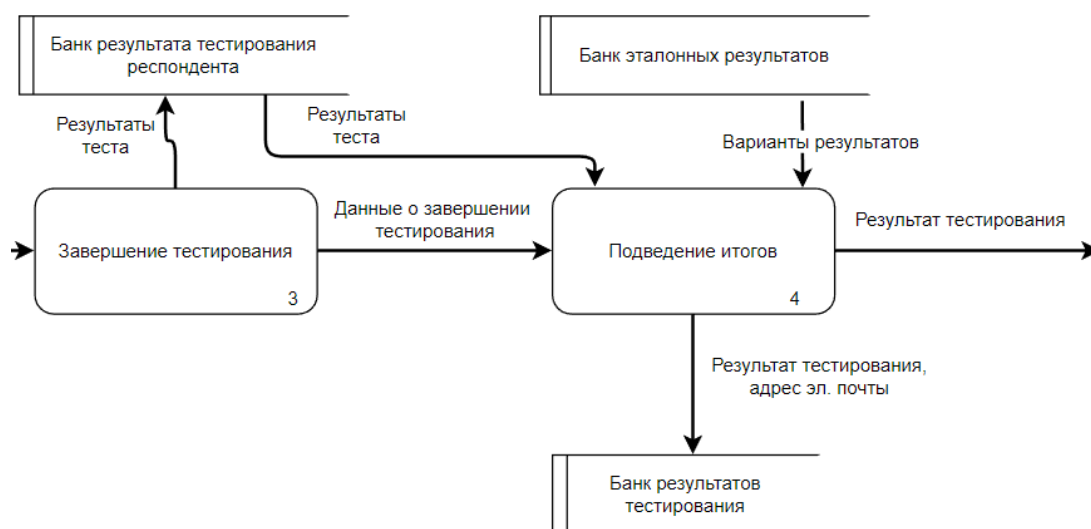


Рисунок 1.15, лист 2

На данной декомпозиции процесса видно, что перед началом прохождения тестирования респондент заполняет анкету, в которую вводит своё имя и адрес электронной почты. Введённые респондентом данные сохраняются в банк респондентов. После заполнения анкеты респондент переходит к процессу «Прохождение тестирования», то есть начинает отвечать на вопросы методики профориентационного тестирования. Вопросы для тестирования подгружаются из «Банка вопросов теста» в соответствии с выбранной методикой тестирования. Каждый вопрос подразумевает наличие некоторого количества вариантов ответов, которые также хранятся в «Банке вопросов». Респондент, отвечая на вопрос, выбирает один из предложенных вариантов ответов. В зависимости от выбранного варианта ответа респонденту начисляется определённое количество баллов. Данный процесс происходит повторно до тех пор, пока респондент не ответит на все вопросы методики тестирования. Ответив на все вопросы, респондент завершает процесс тестирования. После выполняется подсчёт количества баллов, полученный результат сохраняется в «Банк результата тестирования респондента». Следующий процесс, который выполняется, «Подведение итогов». В данном процессе происходит сравнительный анализ результатов, а именно результата, полученного респондентом, с эталонными результатами, которые хранятся в «Банке эталонных результатов». По завершении анализа, в

«Банк результатов тестирования» передаются данные о результате тестирования респондента, который может быть представлен, например, в виде перечня профессий, и адрес электронной почты респондента. По завершении данного процесса респонденту выводится его результат тестирования, с которым он может ознакомиться.

Таким образом, было выполнено моделирование бизнес-процесса, с помощью которого была рассмотрена более подробно концепция веб-приложения профориентационного тестирования, а также рассмотрен предполагаемый функционал.

1.6.4 Модель «сущность-связь»

Для базы данных, в которой будет храниться информация о методиках тестирования и респондентах, прошедших тестирование, было выполнено моделирование *ER*-диаграммы (рис. 1.16).

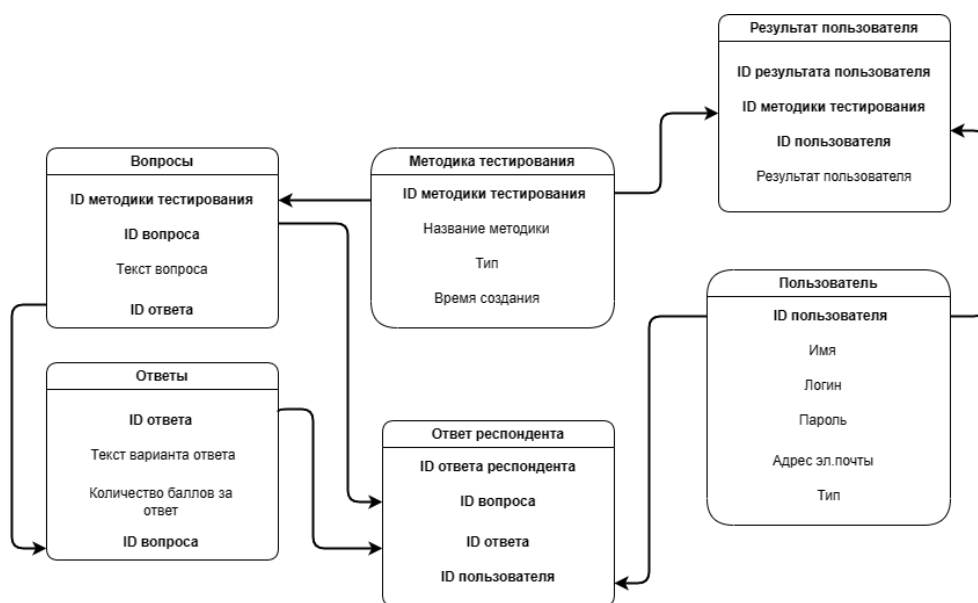


Рисунок 1.16 – *ER*-диаграмма

Сущность «*Методика тестирования*» хранит в себе следующие данные о тесте, который создан в системе: «*ID методики тестирования*», «*Название ме-*

методики», «Тип» и «Время создания». Методика тестирования может быть трёх типов: активная, неопубликованная и закрытая. В зависимости от того в каком состоянии находится методика тестирования, изменяется значение в поле «Тип».

Сущность «Вопросы» хранит в себе вопросы на все методики тестирования, которые представлены в веб приложении. Каждый вопрос содержит в себе несколько вариантов ответов, из которых респондент выбирает, отвечая на данные вопросы в ходе тестирования. Таким образом, между сущностями «Методика тестирования» и была установлена связь **один-ко-многим** через атрибут «ID методики тестирования», а между сущностями «Вопросы» и «Ответы» установлена связь **один-ко-многим** через атрибут «ID вопроса».

Сущность «Пользователь» хранит в себе данные о пользователе, который зарегистрирован в системе. Данные сохраняются в данную таблицу после того, как пользователь заполнил соответствующую форму. Система предполагает три типа пользователей: *super*-пользователь, сотрудник ОДПиНН и респондент. По умолчанию все регистрирующиеся пользователи являются респондентами. Для изменения роли в системе необходимо изменить в базе данных значение поля «Тип» для соответствующего пользователя.

Сущность «Ответ пользователя» является связующей таблицей для трёх сущностей: «Вопросы», «Ответы» и «Пользователь». Данная таблица нужна, чтобы зафиксировать, какой вариант ответа выбрал респондент, отвечая на тот или иной вопрос, чтобы в дальнейшем программа могла вычислить результат тестирования респондента и вывести полученный результат на экран пользователя.

Сущность «Результат пользователя» хранит результат, который получил пользователь, пройдя ту или иную методику тестирования. Данная сущность связана с сущностями «Методика тестирования» и «Пользователь».

Таким образом, все таблицы находятся в третьей нормальной форме, так как в них каждый неключевой атрибут зависит от первичного ключа и отсутствуют транзитивные зависимости.

1.7 Сравнительный анализ и выбор средств разработки веб-приложения «ПрофТест ХТИ»

Веб-приложение является наиболее подходящим вариантом для реализации данного проекта, так как основным преимуществом веб-платформы является кроссплатформенность. Таким образом, приложение будет работать на всех существующих системах независимо от вида устройства, а также операционной системы.

В отличие от других систем, веб-приложение не нуждается в установке на устройство для дальнейшего его использования. Для того чтобы пользователь мог начать работу в приложении, ему достаточно всего лишь перейти по ссылке. Веб-приложение – это клиент-серверное приложение, в котором клиент использует браузер для взаимодействия с веб-сервером. Логика веб-приложения распределяется между сервером и клиентом, данные в основном хранятся на сервере, а обмен информацией осуществляется по сети. Одним из преимуществ этого метода является то, что клиент не зависит от операционной системы конкретного пользователя, поэтому веб-приложение является кроссплатформенным сервисом [14].

1.7.1 Выбор системы управления базами данных

При выборе подходящей системы управления базами данных (СУБД) важно учитывать различные факторы, такие как производительность, масштабируемость, надежность, удобство использования и особенности каждой СУБД. Для анализа были выбраны три популярные СУБД: *MySQL*, *SQLite* и *PostgreSQL* (таблица 1).

Таблица 1 – Сравнительный анализ СУБД

Критерий	MySQL	SQLite	PostgreSQL
Производительность	Предоставляет высокую производительность и масштабируемость для больших баз данных и высоконагруженных систем. Он поддерживает распределенные вычисления и репликацию.	Обладает высокой производительностью благодаря своей встроенной природе. Он не требует отдельного сервера и обеспечивает быстрый доступ к данным.	Обеспечивает хорошую производительность и масштабируемость. Он поддерживает сложные запросы, индексы и транзакции.
Удобство использования	Имеет широкую поддержку и большое сообщество пользователей. Предоставляет разнообразные инструменты и документацию.	Прост в использовании и не требует сложной настройки. Представляет собой один файл, который легко интегрировать в проекты.	Обладает удобством использования и широким функционалом. Поддерживает расширения и предоставляет мощные инструменты для администрирования.
Надежность	Обладает надежностью и обеспечивает сохранность данных с помощью репликации и резервного копирования.	Обеспечивает надежность данных с помощью транзакций и механизма сохранности. Он предоставляет средства для обработки сбоев и восстановления данных.	Предоставляет надежность и целостность данных с помощью транзакций, механизма сохранности и проверки целостности.

В заключение, при разработке профориентационного тестирования лучшим выбором среди рассмотренных СУБД является *SQLite*. Она обладает рядом преимуществ, которые делают ее оптимальным решением для данного типа приложения. *SQLite* обеспечивает высокую производительность, простоту использования и надежность, что особенно важно для проектов малого и среднего масштаба, включая системы профориентационного тестирования.

SQLite предлагает компактность, легкость интеграции и отсутствие необходимости в отдельном сервере, что упрощает развертывание и обслуживание приложения. Также *SQLite* обеспечивает быстрый доступ к данным и эффективное управление ими, что важно для обработки большого объема информации, характерного для профориентационных тестов.

Хотя *MySQL* и *PostgreSQL* также предлагают широкий функционал и масштабируемость, они имеют большую сложность настройки и требуют наличия отдельного сервера. В случае профориентационного тестирования, где

необходима простота использования и ограниченные ресурсы, эти факторы могут быть излишними.

Таким образом, на основе анализа рассмотренных СУБД, рекомендуется использовать *SQLite* при разработке профориентационного тестирования. Он обеспечит надежное и эффективное хранение данных, а также упростит процесс развертывания и обслуживания приложения.

1.7.2 Выбор языков программирования

В настоящее время существует огромное количество языков программирования, однако нет такого языка, который бы одинаково хорошо подходил для решения любой задачи. Поэтому необходимо делать осознанный выбор, исходя из собственных задач, которые предстоит решить.

Если рассматривать направление веб-разработки, то список вариантов не велик, поскольку в основном все браузеры поддерживают одинаковую спецификацию.

HTML (HyperText Markup Language) используется при разработке интерфейса веб-приложения для создания структуры и содержимого веб-страницы. Он определяет разметку элементов и текста, позволяет задавать заголовки, параграфы, списки, таблицы, изображения и другие элементы, которые пользователь видит на странице.

CSS (Cascading Style Sheets) применяется для стилизации и внешнего оформления веб-страницы. С помощью *CSS* можно управлять цветами, шрифтами, размерами, отступами, расположением элементов и другими аспектами визуального представления. Он позволяет создавать современные и привлекательные дизайны, делая веб-страницу более эстетичной и функциональной.

JavaScript – это язык программирования, который добавляет интерактивность и динамическое поведение на веб-странице. Он позволяет обрабатывать события, взаимодействовать с пользователем, выполнять анимацию, валидацию

форм, отправку данных на сервер и другие действия, делая веб-приложение более отзывчивым и интерактивным.

Все три языка – *HTML*, *CSS* и *JavaScript* – взаимодействуют друг с другом для создания полноценного и функционального интерфейса веб-приложения. *HTML* определяет структуру содержимого, *CSS* задает его внешний вид, а *JavaScript* добавляет динамическое поведение и взаимодействие с пользователем. Вместе они обеспечивают создание современных и привлекательных веб-интерфейсов, которые обеспечивают удобство использования и уникальный пользовательский опыт.

Далее необходимо выбрать язык программирования для разработки серверной части веб-приложения. Наиболее популярными вариантами являются *PHP*, *JavaScript*, *Python*.

PHP – это скриптовый язык программирования, который часто используется для создания динамических веб-страниц. Он обладает широкой поддержкой и популярен в веб-разработке. *PHP* предлагает различные фреймворки, такие как *Laravel* и *Symfony*, которые облегчают процесс разработки веб-приложений. Данный язык программирования не подходит по следующим причинам:

- Ограниченные возможности масштабирования: *PHP* в основном используется для создания динамических веб-страниц и имеет свои ограничения в масштабируемости. Так как требуется разработать профориентационное приложение с высокой производительностью, обработкой большого количества пользователей и расширяемостью, другие языки программирования могут предложить более подходящие инструменты и фреймворки.

- Сложность организации и поддержки кода: *PHP* не всегда обладает четкой организацией и структурой кода. Он часто используется для быстрого создания простых веб-страниц, но может быть сложным в поддержке и разработке более сложных приложений. Другие языки программирования, такие как *Python*, могут предложить более структурированный и поддерживаемый подход к разработке веб-приложений.

– Недостаточные инструменты для тестирования и отладки: *PHP* имеет ограниченные инструменты для автоматизированного тестирования и отладки приложений. Другие языки программирования, такие как *Python* или *JavaScript*, предлагают более широкий спектр инструментов и фреймворков для тестирования кода, обнаружения ошибок и отладки.

Исходя из вышеперечисленных особенностей *PHP*, было принято решение выбрать другой язык программирования для разработки.

JavaScript – язык программирования, который широко используется для создания интерактивных веб-страниц и клиентской части веб-приложений. Он предоставляет мощные возможности для работы с браузером и взаимодействия с пользователем. Данный язык программирования не подходит для данного проекта по следующим причинам:

– Ограниченные возможности серверной разработки: *JavaScript* в основном используется в браузерной среде для создания интерактивных веб-страниц. В отличие от других языков программирования, таких как *PHP* и *Python*, *JavaScript* не является первоначально предназначенным для серверной разработки. Хотя с помощью *Node.js* можно создавать серверные приложения на *JavaScript*, это не является его основным предназначением, и у него могут быть ограниченные возможности и инструменты для разработки серверной логики.

– Сложность масштабирования: *JavaScript*, особенно в контексте *Node.js*, может столкнуться с ограничениями масштабирования при обработке большого количества одновременных запросов или при работе с большими объемами данных. Другие языки программирования и фреймворки, такие как *Python*, предлагают более эффективные и масштабируемые инструменты для разработки серверной части приложений.

– Ограниченные инструменты и фреймворки: *JavaScript* имеет некоторые фреймворки для серверной разработки, такие как *Express.js* или *Nest.js*, однако их возможности и экосистема не так широки и развиты, как в случае с другими языками программирования. Использование языка программирования

с более развитыми инструментами и фреймворками может упростить разработку и обеспечить более гибкую и эффективную серверную логику для профориентационного тестирования.

– Ограниченный доступ к базам данных: *JavaScript* не обладает богатым выбором драйверов и библиотек для работы с различными типами баз данных, особенно с реляционными базами данных. Использование языка программирования, который имеет более широкую поддержку баз данных, может облегчить взаимодействие с базой данных и обеспечить более надежное хранение и доступ к данным профориентационного тестирования.

Таким образом, *JavaScript* может быть использован для разработки серверной части, но альтернативный язык программирования такой, как *Python*, может предоставить более подходящие и эффективные решения в данном контексте.

Python – универсальный язык программирования, который нашел широкое применение в различных областях, включая веб-разработку. Данный язык программирования имеет ряд преимуществ. Например, такие как:

– Простота и выразительность: *Python* известен своей простотой чтения и понимания, что делает его идеальным выбором для разработки веб-приложений. Его чистый и лаконичный синтаксис позволяет разработчикам быстро писать и поддерживать код. *Python* также обладает широким спектром библиотек и фреймворков, которые упрощают разработку серверной логики и предоставляют мощные инструменты для обработки данных и взаимодействия с базами данных.

– Мощные фреймворки: *Python* имеет несколько популярных фреймворков для разработки веб-приложений, таких как *Django* и *Flask*. Фреймворки *Python* предлагают множество готовых инструментов и функций, упрощающих создание серверной логики, маршрутизацию, управление базами данных и обработку запросов.

– Широкая поддержка баз данных: *Python* имеет разнообразные библиотеки для взаимодействия с различными типами баз данных, включая реля-

ционные базы данных, такие как *MySQL*, *PostgreSQL*, *SQLite*, а также *NoSQL*-базы данных, такие как *MongoDB*. Это позволяет разработчикам выбрать наиболее подходящую базу данных для их приложения профориентационного тестирования и эффективно взаимодействовать с ней через *Python*.

– Широкие возможности интеграции: *Python* предлагает множество инструментов и библиотек для интеграции с другими системами и сервисами. Это может быть полезно для расширения функциональности приложения профориентационного тестирования, взаимодействия с внешними *API* или интеграции с другими приложениями.

Особое внимание следует уделить фреймворкам *Django* и *Flask* для разработки веб-приложений на языке *Python*.

Django – это полнофункциональный фреймворк, разработанный для быстрой и эффективной разработки веб-приложений. Он предоставляет готовые решения для обработки запросов, аутентификации, управления базами данных и административной панели. *Django* позволяет разработчикам создавать сложные проекты с минимальными усилиями благодаря встроенным компонентам и конвенциям разработки [15].

Flask – легкий и гибкий фреймворк для разработки веб-приложений на языке *Python*. Он предоставляет базовый набор инструментов и оставляет большую свободу выбора в проектировании приложения. *Flask* отличается простотой использования и является отличным выбором для небольших и средних проектов, где требуется гибкость и минимальные ограничения [16].

В таблице 2 представлен более подробный анализ данных фреймворков.

Таблица 2 – Сравнительный анализ фреймворков

Критерий	Django	Flask
Функциональность	Предлагает более широкий набор функций и включает множество инструментов из коробки.	Оставляет больше свободы выбора и позволяет разработчикам выбирать только необходимые компоненты.
Сложность	Имеет более высокий порог входа из-за своей полноты и сложности.	Является более простым и легким в освоении.

Окончание таблицы 2

Масштабируемость	Обеспечивает хорошую масштабируемость и может использоваться для разработки крупных проектов.	Ориентирован на небольшие и средние проекты, но также может масштабироваться с использованием дополнительных расширений.
Гибкость	Предлагает более жесткую структуру и конвенции разработки.	Предоставляет большую гибкость в выборе архитектуры и компонентов приложения.

Учитывая все вышеперечисленные факторы, использование языка программирования *Python* для разработки серверной части веб-приложения представляется наилучшим выбором в настоящее время. Среди фреймворков для *Python*, *Flask* отличается простотой и гибкостью, что делает его предпочтительным решением для небольших и средних проектов, где важны легкость разработки и минимальные ограничения.

Таким образом, для разработки интерфейса веб-приложения будут использоваться *HTML*, *CSS* и *JavaScript*. Для построения логики серверной части приложения будет использоваться фреймворк *Flask*.

Вывод по разделу «Анализ предметной области»

В результате анализа предметной области было выявлено, что все мероприятия, проводимые сотрудниками ОДПиНН и сотрудниками ЦПЮИ, играют важную роль для привлечения будущих абитуриентов. Также была выявлена проблема выбора направления подготовки абитуриентами при поступлении в высшее учебное заведение. Для решения данной проблем будет использоваться создаваемая информационная система, которая позволит пройти профориентационное тестирование по четырём методикам. Данная ИС поможет респонденту определиться с выбором будущей профессии и сделать правильный выбор направления подготовки при поступлении в ХТИ – филиал СФУ.

Был выполнен анализ прототипа, который был разработан ранее для ХТИ – филиала СФУ, а также проведён анализ аналогичных решений, которые пред-

ставлены в открытом доступе. Прототип не может использоваться в дальнейшей работе приёмной кампании по причине несоответствия ИС современным требованиям. Сервис «Тестометрика» и сервис «Работа России» не могут использоваться, так как данные сервисы не предоставляют необходимую настройку доступа к материалам. Таким образом, было обосновано решение разработки собственной ИС для института.

Исходя из требований заказчика, были спроектированы модели:

- модель бизнес-процесса, отображающая структуру и функции системы «Как будет»;
- модель действий пользователя в информационной системе (*IDEF3*);
- модель потока данных, отражающая функциональные процессы, связанные с потоками данных (*DFD*).

На их основе был определён основной функционал информационной системы профориентационного тестирования.

Исходя из предъявляемых требований, были определены необходимые инструменты для реализации проекта. Для управления базами данных был выбран *SQLite*, в то время как для разработки пользовательского интерфейса были выбраны *HTML*, *CSS* и *JavaScript*. Для реализации серверной части был выбран фреймворк *Flask*.

2 Описание процесса разработки информационной системы профориентационного тестирования

2.1 Настройка среды разработки

Настройка среды разработки перед началом работы над созданием веб-приложения важна, потому что она обеспечивает эффективную и продуктивную работу над созданием нового проекта. Правильно настроенная среда разработки позволяет минимизировать возможные проблемы и ошибки в процессе разработки.

Так как для разработки серверной части приложения было принято решение использовать фреймворк *Flask*, первым делом необходимо скачать установщик *Python* с официального сайта. Для разработки использовалась последняя версия 3.11.4. Далее запустить установщик и, следовать инструкциям по установке *Python*. Рекомендуется выбрать опцию «*Add Python to PATH*» во время установки, чтобы иметь возможность запускать *Python* из командной строки.

Далее для работы необходимо установить редактор кода, внутри которого будет выполняться процесс разработки программного продукта. Для работы был выбран *Visual Studio Code*. Основной причиной использования является его мощная и гибкая среда разработки, которая обеспечивает широкий набор инструментов и функций для написания, отладки и тестирования кода, ускоряя процесс разработки и повышая производительность разработчиков. Кроме того, *Visual Studio Code* поддерживает широкий спектр языков программирования, включая *HTML*, *CSS*, *JavaScript* и многие другие, а также предоставляет возможность установки дополнительных расширений и плагинов, что делает его идеальным инструментом для разработки веб-приложений любого уровня сложности и масштаба. Для установки необходимо перейти на официальный сайт и скачать установщик.

Далее необходимо открыть редактор кода и через терминал установить модули *flask* и *sqlalchemy*. Для установки модулей необходимо использовать команду `pip install flask sqlalchemy`.

Далее необходимо создать папку для проекта. В данной папке необходимо создать файл, в котором будет прописан код для создания экземпляра приложения *Flask* и настройки базы данных *SQLAlchemy* (рисунок 2.1).

```
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3 |
4 app = Flask(__name__)
5
6 app.secret_key = "*****"
7 app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///pychart.db"
8 app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
9
10 db = SQLAlchemy(app)
```

Рисунок 2.1 – Код программы файла *server.py*

Для обеспечения резервного копирования проекта было решено использовать системы контроля версий *Git*. Это позволит легко управлять версиями кода на протяжении всего процесса разработки и иметь резервные копии проекта на случай непредвиденных ситуаций. Для этого необходимо создать закрытый репозиторий и загрузить на него текущую версию проекта. В дальнейшем достаточно будет создавать *commit* и выгружать только тех файлы, которые были созданы либо изменены в процессе работы. Данный подход экономит время на создание резервной копии проекта, а также позволяет отслеживать изменения, внесённые в проект на протяжении всей работы.

На этом этапе подготовительные работы для настройки среды разработки были завершены. Далее необходимо разработать модель базы данных и настроить работу с ней внутри приложения.

2.2 Разработка базы данных

Внутри проекта был создан файл *data_models.py*. В данном файле представлен код, который описывает структуру базы данных. Сначала импортируются необходимые модули: *ast* и *datetime*, а затем импортируется объект *db* из модуля *server*.

Далее определены классы *User*, *Question*, *Survey*, *Response*, *Statistics* и *Sender*, которые являются моделями базы данных, использующими объект *db.Model*. Каждый класс представляет отдельную таблицу в базе данных, и поля каждого класса определяют столбцы в соответствующей таблице.

Класс *User* отвечает за создание таблицы «*USERS*», в которой будут храниться данные о зарегистрированных пользователях системы.

Класс *Question* отвечает за создание таблицы «*QUESTIONS*», в которой будут храниться данные вопросов, созданных в системе.

Класс *Survey* отвечает за создание таблицы «*SURVEYS*», в которой будут храниться данные созданных в системе тестов на основе методик профориентационного тестирования либо опросов для обратной связи с пользователями.

Класс *Response* отвечает за создание таблицы «*RESPONSES*», в которой хранятся данные о ответах пользователя на вопросы теста.

Класс *Statistics* отвечает за создание таблицы «*STATISTICS*», в которой будут храниться данные о результатах респондентов, на основе которых будет формироваться статистика.

Класс *Sender* отвечает за создание таблицы «*SENDER*», в которой будут храниться данные о отправителе письма, которое будет получать респондент.

Например, класс *Question* определяет таблицу «*QUESTIONS*» с полями *id*, *text*, *state*, *type*, *required* и *responses* (рис. 2.2). Аналогично, другие классы определяют свои таблицы и соответствующие поля.

```
class Question(db.Model):
    __tablename__ = "QUESTIONS"
    id = db.Column(db.Integer, primary_key=True, nullable=False)
    text = db.Column(db.Text, nullable=False)
    state = db.Column(db.Integer, nullable=False)
    type = db.Column(db.Integer, nullable=False)
    required = db.Column(db.Integer, nullable=False)
    responses = db.Column(db.Text, nullable=False)

    def responsesList(self):
        return ast.literal_eval(self.responses)
```

Рисунок 2.2 – Код модели таблицы *Questions*

В классе *Question* есть метод *responsesList()*, который использует функцию *ast.literal_eval()* для преобразования значения поля *responses* из строки в список. Аналогично, в классе *Survey* есть метод *questionsList()*, который преобразует значение поля *questions* из строки в список.

Эти модели могут быть использованы для создания, чтения, обновления и удаления данных в базе данных, связанной с приложением. *SQLAlchemy* обеспечивает удобный и выразительный способ работы с базами данных в *Python*, а модуль *ast* используется в данном случае для преобразования строковых значений в структуры данных *Python*, чтобы в дальнейшем было удобно взаимодействовать с данными при написании логики расчётов для тестов.

Таким образом, была разработана модель базы данных, а также методы для добавления, обновления и поиска данных в базе данных, которые необходимы для построения дальнейшей логики приложения.

2.3 Разработка формы регистрации и авторизации пользователей в системе

В большинстве современных веб-приложений форма регистрации и авторизации играет важнейшую роль в обеспечении безопасности, конфиденциальности и индивидуализированного опыта для пользователей. Она представляет собой неотъемлемую часть таких приложений, защищая их от несанкционированного доступа. С помощью формы регистрации и авторизации пользователи могут создавать уникальные учетные записи, используя пароли, которые обеспечивают конфиденциальность. Это помогает предотвратить несанкционированный доступ к чувствительным данным и обеспечить защиту пользователей. Также форма авторизации позволяет веб-приложению эффективно управлять доступом пользователей к различным функциям и ресурсам, а форма регистрации собирает дополнительную информацию о пользователях, что может быть полезно для улучшения предоставляемых услуг.

Для системы профориентационного тестирования была разработана следующая форма регистрации, которая представлена на рисунках 2.3 – 2.4.

При регистрации пользователь должен ввести следующие данные:

- логин;
- адрес электронной почты;
- имя пользователя;
- пароль;
- подтверждение пароля.

Кроме того, пользователь обязан дать согласие на обработку персональных данных, так как при регистрации он указывает свой адрес электронной почты.

Регистрация нового пользователя

Логин

Адрес электронной почты Заполните это поле.

Ваше имя

Пароль

Повторите пароль

Согласие на обработку персональных данных *

[Зарегистрироваться](#) [Назад](#)

Рисунок 2.3 – Форма регистрации *desktop*-версия

Регистрация нового пользователя

Логин

Адрес электронной почты Заполните это поле.

Ваше имя

Пароль

Повторите пароль

Согласие на обработку персональных данных *

[Зарегистрироваться](#) [Назад](#)

Рисунок 2.4 – Форма регистрации *mobile*-версия

На рисунке 2.5 представлен код, с помощью которого выполняется регистрация пользователя в системе.

```

def create_user(self, uid, username, password, type, email, real_name):
    # check parameters valid
    if (
        password.isspace()
        or password == ""
        or username.isspace()
        or username == ""
        or email == ""
        or email.isspace()
        or real_name == ""
        or email.isspace()
    ):
        return 0
    if User.query.filter_by(username=username).first():
        return 0
    if User.query.filter_by(id=uid).first():
        return 0
    if type > 4 or type < 1:
        return 0
    if not re.match("[^@]+@[^@]+\.[^@]+", email):
        return 0

    newU = User(
        id=uid,
        username=str.lower(username),
        password=generate_password_hash(password),
        type=type,
        email=email,
        real_name=real_name,
    )
    db.session.add(newU)
    db.session.commit()
    return 1

```

Рисунок 2.5 – Код программы для регистрации пользователя

В начале функции проводится проверка валидности параметров, которые вводит пользователь в форму. В случае, если пользователь не заполнил один из параметров, то функция возвращает ошибку валидности. Также при регистрации необходимо выполнить проверку существующих пользователей в системе. Если пользователь уже зарегистрирован в системе, то функция выдаст соответствующую ошибку. Кроме того, необходимо выполнить проверку корректности введенного адреса электронной почты. Для этого используется регулярное выражение. В случае, если введенный адрес не соответствует шаблону, то пользователь не сможет зарегистрироваться в системе. В случае, если пользователь пройдет все проверки успешно, то создаётся новый объект «*User*» с использованием предоставленных параметров. Пароль хэшируется с помощью функции *generate_password_hash*, чтобы обеспечить безопасное хранение пароля. Новый пользователь добавляется в сеанс базы данных *db.session*.

Кроме того, реализована регистрация пользователя с помощью *CSV*-файла. Для это был создан метод *csv_import* (рис. 2.6), который предназначен для импорта данных из *CSV*-файлов в базу данных.

```
def csv_import(self):
    if not (
        os.path.isfile("data/passwords.csv")
    ):
        return 0 # failure

    with open("data/passwords.csv", "r") as csv_in:
        reader = csv.reader(csv_in)
        for row in reader:
            u_type = 0
            if row[3] == "student":
                u_type = 1
            elif row[3] == "staff":
                u_type = 2

            if not User.query.filter_by(id=row[0]).first() and u_type:
                user = User(
                    id=row[0],
                    username=row[1],
                    password=generate_password_hash(row[2]),
                    type=u_type,
                    email=row[4],
                    real_name=row[5],
                )
                db.session.add(user)

    db.session.commit()

    return 1 # success
```

Рисунок 2.6 – Регистрация пользователя через *csv*-файл

Данная функция проверяет наличие *CSV*-файла *passwords.csv* в директории *data*, в который записываются данные пользователей для регистрации в системе. Таким способом можно зарегистрировать как нового сотрудника ОД-ПиНН, так и респондента. Функция считывает содержимое файла и для каждой строки создает объект *User*, используя данные из строки, такие как идентификатор пользователя, имя пользователя, хешированный пароль, тип пользователя и другие. Если пользователь с таким идентификатором уже существует, запись не добавляется. Функция сохраняет созданных пользователей в базе данных.

Так как система ориентирована на три типа пользователя необходимо реализовать проверку пользователя при авторизации (рис. 2.7).


```

def authenticate(self, user_name, password):
    user = User.query.filter_by(username=str.lower(user_name)).first()

    if not user:
        return False

    if check_password_hash(user.password, password) and user.type > 0:
        session["logged_in"] = True
        session["username"] = user.username
        session["user_id"] = user.id
        session["user_type"] = user.type
        return True

    return False

def check_login(self):
    if session.get("user_type"):
        return session["user_type"]
    else:
        return 0

```

Рисунок 2.7 – Код программы авторизации пользователя

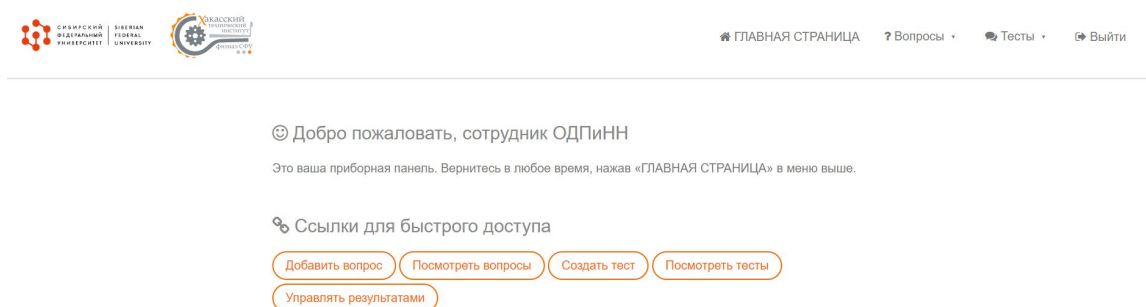
Метод *authenticate* используется для аутентификации пользователя. Данный метод создан для поиска пользователя в базе данных по его имени. Если пользователь найден, то функция возвращает хешированный пароль пользователя (*user.password*) с предоставленным паролем (*password*), используя функцию *check_password_hash* из модуля *werkzeug.security*. Если пароль совпадает и тип пользователя больше 0, сессия пользователя устанавливается на валидную, и функция возвращает *True*, обозначая успешную аутентификацию, в противном случае будет ошибка.

Метод *check_login* используется для проверки типа пользователя, который вошел в систему. Он проверяет наличие значения *user_type* в сеансе (*session*). Если значение существует, оно возвращается. Если значение *user_type* отсутствует в сеансе, функция возвращает 0, обозначая, что пользователь не вошел в систему или сеанс истек.

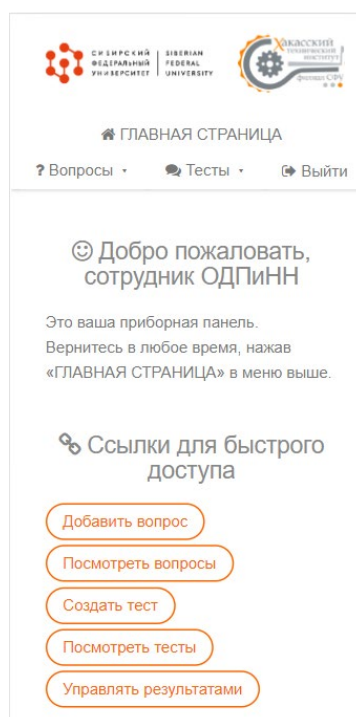
Таким образом, была разработана логика регистрации и авторизации пользователей в системе, а также настроен процесс сохранения данных новых пользователей в базу данных информационной системы.

2.3 Разработка личного кабинета сотрудника ОДПиНН

Данный пользователь должен иметь возможность выполнять ряд функций, которые были описаны в п.1.6.2. Для него была разработана панель управления, с помощью которой данный пользователь выполнять все свои функции (рис. 2.8 – 2.9).



Рисунки 2.8 – Панель управления сотрудника ОДПиНН *desktop*-версия



Рисунки 2.9 – Панель управления сотрудника ОДПиНН *mobile*-версия

Первый шаг в разработке заключается в создании функции для добавления вопросов в систему, чтобы составлять тесты для профориентационного тестирования. Для этой цели была разработана форма (рис. 2.10), которая включает кнопки для автоматического ввода нескольких вариантов ответов. Это помогает автоматизировать процесс создания вопросов для определенных методик профориентационного тестирования. Кроме того, в систему внедрена возможность разделения вопросов на обязательные и необязательные, а также предоставлено текстовое поле для ввода ответа, если требуется создание опросов для обратной связи с пользователями.

СФЕРАСЕРВИС | СИСТЕМА
ФЕДЕРАЛЬНЫЙ ЦЕНТРАЛЬНЫЙ
УНИВЕРСИТЕТ

УЛЬЯНОВСКИЙ
УНИВЕРСИТЕТ

ГЛАВНАЯ СТРАНИЦА ? Вопросы Тесты Выйти

+ Добавить новый вопрос

Текст вопроса

Тип вопроса Необязательный Обязательный

Тип ответа Несколько вариантов Текст

Ответ 1

Ответ 2

+ Да/Нет Шкала Лайкерта

Шкала Лайкерта (4) Сброс

Добавить вопрос

Рисунок 2.10 – Форма для создания вопроса

В случае, если сотруднику нужно удалить или отредактировать вопросы, на панели управления ему необходимо нажать на кнопку «Посмотреть вопросы», чтобы перейти на страницу, где будет представлен список вопросов, которые он создал. Просмотреть список и удалить или отредактировать необходимые вопросы, нажав на соответствующую кнопку (рис. 2.11).

При редактировании вопроса пользователю открывается форма для редактирования, с помощью которой он может внести изменение в текст вопроса или варианты ответов и сохранить изменения в базу данных (рис. 2.12).

* Список вопросов

Вопрос

Хотелось бы изучать разнообразие животного и растительного мира?

Тип

Обязательный

Ответы

1. Очень хотелось бы, нравится этим заниматься // 2. Нравится, но не очень сильно // 3. Затрудняюсь ответить // 4. Не хотелось бы, не нравится // 5. Очень не хотелось бы, совершенно не нравится



Вопрос

Интересно ли проводить физические эксперименты?

Тип

Обязательный

Ответы

1. Очень хотелось бы, нравится этим заниматься // 2. Нравится, но не очень сильно // 3. Затрудняюсь

Рисунок 2.11 – Список созданных вопросов в системе

✎ Редактировать вопрос

Текст вопроса

Тип вопроса Необязательный Обязательный

Тип ответа Несколько вариантов Текст

Ответ 1

Ответ 2

Ответ 3

Ответ 4

Ответ 5

Рисунок 2.12 – Форма редактирования вопроса

После создания всех необходимых вопросов для методики профориентационного тестирования специалист должен на их основе создать тест в системе. Для этого он выбирает на панели управления кнопку «Создать тест» и переходит на страницу, где представлен список всех вопросов, хранящихся в системе (рис. 2.13).

+ Добавить новый тест

Название теста

Введите название теста

Заполните это поле.

Выделить все

Создать тест

Выберите вопросы, которые вы хотите включить.

- Хотелось бы изучать разнообразие животного и растительного мира? (Техт)
- Интересно ли проводить физические эксперименты? (Техт)
- Хотелось бы разрабатывать новые технологии химического синтеза? (Техт)

Рисунок 2.13 – Создание нового теста

Далее представлен код, который отвечает за создание нового теста в системе (рис. 2.14).

```

def create_survey(self, name, questions):
    if name.isspace() or name == "" or not questions:
        return 0
    newS = Survey(name=name, state=1, questions=str(questions))
    db.session.add(newS)
    db.session.commit()
    return 1
    
```

Рисунок 2.14 – Код программы для создание нового теста в системе

Данный код выполняет проверку, заполнено ли поле «Наименование теста», а также выбраны ли вопросы из представленного списка. В случае, если одно из условий не выполнено, то тест не будет сохранён в системе.

После создания теста сотрудник ОДПиНН оповещает *super*-пользователя, чтобы тот, если это необходимо, доработал систему и опубликовал созданный тест, чтобы респонденты могли пойти тестирование.

Кроме того, сотрудник ОДПиНН имеет возможность просматривать результаты тестов, а также закрывать их на время. Для этого ему необходимо нажать на панели управления кнопку «*Посмотреть тесты*» и перейти на соответствующую страницу (рис. 2.15 – 2.16).

⌚ Необработанные тесты

Название теста

Тест

Создан

04/06/2023 22:57

[Закреть](#)

👁️ Активные тесты

Название теста

Карта интересов - А.Е.Голомшток

Создан

01/06/2023 17:54

[Посмотреть результаты](#)

[Закреть](#)

Рисунок 2.15 – Страница просмотра тестов *desktop*-версия

⌚ Необработанные тесты

Название теста

Тест

Создан

04/06/2023 22:57

[Закреть](#)

👁️ Активные тесты

Название теста

Карта интересов - А.Е.Голомшток

Создан

01/06/2023 17:54

[Посмотреть результаты](#)

[Закреть](#)

Рисунок 2.16 – Страница просмотра тестов *mobile*-версия

В ходе работы было принято решение визуализировать данные результатов тестов в виде диаграмм. Такой подход обладает рядом преимуществ. Диаграммы позволяют представить большие объемы данных в компактной и наглядной форме. Они помогают увидеть связи, тренды и распределение данных сразу же, без необходимости прочитывать и анализировать длинные текстовые описания или таблицы.

Для построения диаграмм на основе полученных данных тестирования используется интерактивная веб-служба *Google Charts*. Данная библиотека разработана для создания интерактивных и красочных диаграмм и графиков, которые можно встраивать в веб-приложения и веб-страницы. *Google Charts* использует *JavaScript* и *HTML5* для отображения графиков в браузере. Она предоставляет различные типы диаграмм, включая столбчатые, круговые, линейные, гистограммы, спарклайны и др., а также дополнительные возможности, такие как интерактивность, анимация, подписи осей, легенды и пользовательские стили. Для работы необходимо подключить данную библиотеку к странице, на которой будут отображаться графики, а также прописать код, в котором указать параметры для создания диаграммы. Рисунок 2.17 показывает готовый результат построения диаграммы на примере методики профориентационного тестирования Дж. Холланда.



Рисунок 2.17 – Статистика результатов, лист 1

Все результаты

Профессиональный тип личности	Количество совпадений
Реалистический	2
Предприимчивый	1
Артистический	3
Социальный	1

Рисунок 2.17, лист 2

Кроме того, сотрудник может посмотреть результат отдельного пользователя. Для этого ему необходимо выбрать на панели управления «Управлять результатами» и перейти на страницу, где будет представлен список пользователей. Для каждого пользователя указан свой перечень методик профориентационного тестирования, по которым он прошёл тестирование в системе. Таким образом, сотрудник обладает возможностью просмотра результата, который выводится на экран респондента по завершении тестирования. На рисунке 2.18 представлена страница «Управление результатами».

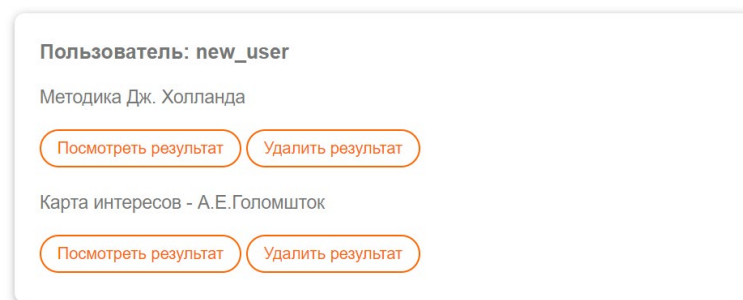


Рисунок 2.18 – Результаты тестирования пользователя

Далее представлен код программы для проверки результата респондента.


```

@app.route("/get_user_responses/<survey_id>/<user_id>")
def get_user_responses(survey_id, user_id):
    user_responses = Response.query.filter_by(u_id=user_id, s_id=survey_id).all()

    if len(user_responses) == 0:
        return "Для данного опроса и пользователя результатов не найдено."

    responses_data = []
    for response in user_responses:
        response_data = {
            "question_id": response.q_id,
            "question_text": response.question.text,
            "response_text": response.text,
            "response_num": response.num,
        }
        responses_data.append(response_data)

    return render_template("/get_user_responses.html", responses_data=responses_data)

```

Рисунок 2.19 – Проверка наличия результата пользователя

При обращении к этому маршруту, код сначала выполняет запрос к базе данных, используя модель «*Response*». Он ищет все записи, где идентификатор пользователя соответствует указанному *user_id*, а идентификатор опроса соответствует указанному *survey_id*. Результаты запроса сохраняются в переменную *user_responses*.

Затем код проверяет количество полученных результатов. Если результатов нет, то возвращается сообщение «Для данного опроса и пользователя результатов не найдено».

Если есть результаты, то код проходит по каждому объекту *response* в *user_responses* и создает словарь *response_data*, содержащий информацию о вопросе и ответе пользователя. В этом словаре сохраняются идентификатор вопроса, текст вопроса, текст ответа и числовой ответ, если такие имеются. Словари *response_data* добавляются в список *responses_data*.

Наконец, код возвращает шаблон «*/get_user_responses.html*» с передачей списка *responses_data* в качестве данных для отображения результатов пользователю. Этот шаблон может использоваться для отображения результатов опроса для конкретного пользователя на веб-странице.

Таким образом, был разработан личный кабинет для сотрудника ОД-ПиНН. Описан процесс разработки функций для данного пользователя, а также выполнено описание процесса работы данного пользователя в системе.

2.4 Разработка личного кабинета *super*-пользователя

Данный пользователь ответственен за обслуживание информационной системы. В его обязанности входит устранение ошибок, которые могут возникнуть в процессе эксплуатации, а также доработка системы в случае, если будет принято решение о внедрении дополнительных методик профориентационного тестирования. Так как работа данного пользователя несёт значительный вклад в процесс создания профориентационного теста внутри системы, для него необходимо разработать отдельный личный кабинет.

Основной функцией в рамках системы данного пользователя является публикация теста. После того, как сотрудник ОДПиНН добавил в систему вопросы и создал на их основе новый тест, на панели управления *super*-пользователя появляется созданный тест, который имеет статус «*Необработанный*» (рис. 2.20 –2.21).

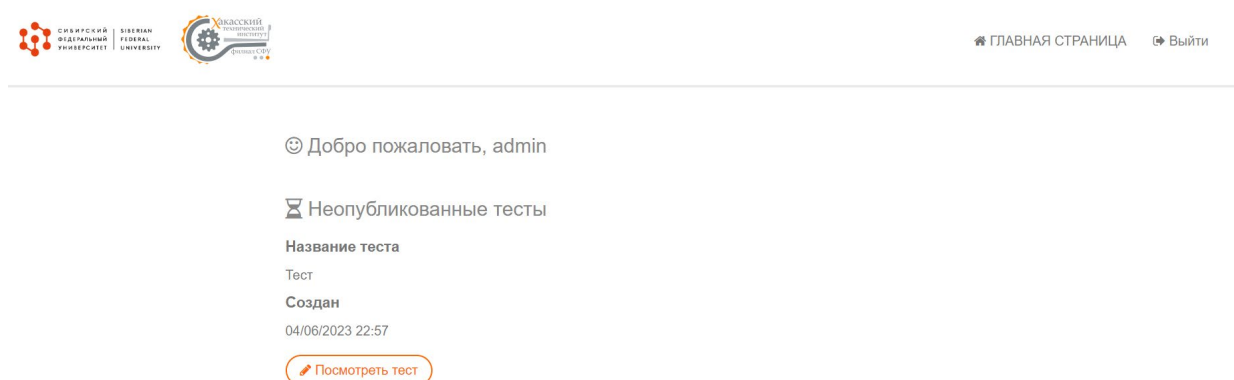


Рисунок 2.20 – Панель управления *super*-пользователя *desktop*-версия



Рисунок 2.21 – Панель управления *super*-пользователя *mobile*-версия

Если созданный тест удовлетворяет функционалу информационной системы, то пользователь может сразу опубликовать его. Для этого ему необходимо открыть тест и нажать на кнопку «*Опубликовать тест*» (рис. 2.22).

```
if system.update_survey(sid, survey.name, survey.c_id, 2, s_questions):
    return render_template(
        "review.html",
        survey=survey,
        questions=questions,
        optional_questions=optional_questions,
        success=1,
    )
else:
    return render_template(
        "review.html",
        survey=survey,
        questions=questions,
        optional_questions=optional_questions,
        error=0,
    )
else:
    return render_template(
        "review.html",
        survey=survey,
        questions=questions,
        optional_questions=optional_questions,
    )
```

Рисунок 2.22 – Код программы для публикации теста в системе

Так как профориентационные методики тестирования не подразумевают разбиение ответов на правильные и неправильные, для них нельзя написать универсальный алгоритм подсчёта результата. В настоящий момент пользователь может без внесения изменений в систему опубликовать только опросы для обратной связи с пользователями, созданные сотрудником ОДПиНН. Для внедрения новых методик профориентационного тестирования пользователю необходимо перед публикацией теста разработать для него новую логику для теста в соответствии с методикой тестирования. Программный код обработки результатов для выбранных методик тестирования приведен в приложении А.

Таким образом, был разработан личный кабинет для *super*-пользователя в соответствии с его основной функцией, а также описан процесс работы данного пользователя в системе.

2.4 Разработка личного кабинета респондента

Так как основной задачей пользователя является прохождение теста, необходимо реализовать личный кабинет таким образом, чтобы респонденту было удобно взаимодействовать с системой в процессе прохождения тестирования и в работе в целом.

На панели управления пользователя отображается список методик профориентационного тестирования, которые доступны ему в текущий момент времени (рис. 2.23 – 2.24). Для того чтобы пройти тест, пользователю необходимо выбрать методику тестирования и нажать на кнопку «*Открыть тест*» и перейти на страницу с тестом.

😊 Добро пожаловать, new_user

👁️ Открытые тесты

Карта интересов - А.Е.Голомшток

01/06/2023 17:54

🔗 Открыть тест

👁️ Посмотреть результаты

Рисунок 2.23 – Панель управления респондента *desktop*-версия



😊 Добро пожаловать, 100

👁️ Открытые тесты

Карта интересов -
А.Е.Голомшток

01/06/2023 17:54

🔗 Открыть тест

👁️ Посмотреть результаты

Рисунок 2.24 – Панель управления респондента *mobile*-версия

В процессе прохождения тестирования респондент обязан ответить на все вопросы методики профорientационного тестирования, чтобы получить

наиболее достоверный результат. Исходя из этого, была разработана функция, которая проверяет наличие ответов на обязательные вопросы.

```
{% if question.required != 1 %}
<input name="{{ question.id }}" type="radio" value="{{ loop.index }}" class="radio-input custom-checkbox check-button">
  {{ response }}
</label>
{% else %}
<input name="{{ question.id }}" type="radio" value="{{ loop.index }}" class="radio-input custom-checkbox check-button" required>
  {{ response }}
</label>
{% endif %}
{% endfor %}
<br>
{% else %}
<textarea class="pure-input-1-2" name="{{ question.id }}"
  placeholder="Пожалуйста, введите свой ответ..."></textarea><br>
{% endif %}
{% endfor %}
<button type="submit" name="bt" class="pure-button pure-button-primary">Завершить
  тестирование</button>
```

Рисунок 2.25 – Код программы для проверки пропущенных вопросов теста

Карта интересов - А.Е.Голомшток 🏠 ГЛАВНАЯ СТРАНИЦА 👤 Выйти

🗨️ Пожалуйста, пройдите тест, представленный ниже.

1. Хотелось бы изучать разнообразие животного и растительного мира? *

Очень хотелось бы, нравится этим заниматься
 Нравится, но не очень сильно
 Затрудняюсь ответить
 Не хотелось бы, не нравится
 Очень не хотелось бы, совершенно не нравится

2. Интересно ли проводить физические эксперименты? *

Очень хотелось бы, нравится этим заниматься
 Нравится, но не очень сильно
 Затрудняюсь ответить
 Не хотелось бы, не нравится
 Очень не хотелось бы, совершенно не нравится

3. Хотелось бы разрабатывать новые технологии химического синтеза? *

Очень хотелось бы, нравится этим заниматься
 Нравится, но не очень сильно
 Затрудняюсь ответить
 Не хотелось бы, не нравится
 Очень не хотелось бы, совершенно не нравится

⚠️ Выберите один из вариантов.

Рисунок 2.26 – Проверка на наличие пропущенных ответов *desktop*-версия

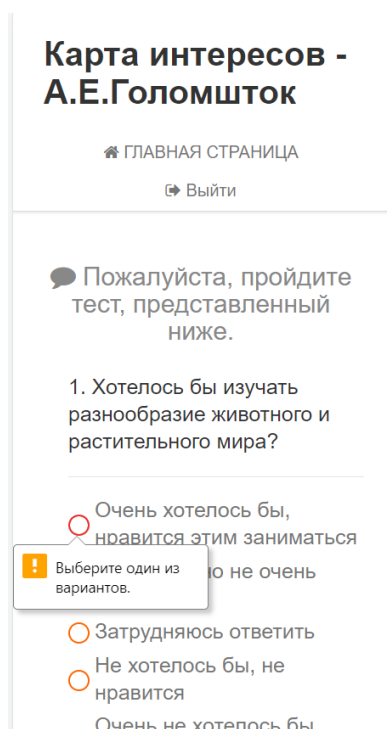


Рисунок 2.27 – Проверка на наличие пропущенных ответов *mobile*-версия

По завершении тестирования пользователю открывается страница результата, где он может ознакомиться со своим результатом. При желании пользователь может сохранить полученный результат к себе на почту, нажав на кнопку «Сохранить результат на почту» (рис. 2.28).

Больше всего вам подходит: **История**

со счётом 10 очков

Все результаты

Биология	3	Интерес выражен слабо
Физика	0	Интереса не вызывает
Химия	-5	Интереса не вызывает
География	-4	Интереса не вызывает
Медицина	8	Ярко выраженный интерес
Техника и электроника	3	Интерес выражен слабо
Строительство	-1	Интереса не вызывает
Математика	2	Интерес выражен слабо
Экономика	3	Интерес выражен слабо
Иностранные языки	2	Интерес выражен слабо
Транспорт	5	Выраженный интерес
Авиация, Морское дело	5	Выраженный интерес
Военные специальности	2	Интерес выражен слабо
История	10	Ярко выраженный интерес
Рабочие специальности	7	Выраженный интерес
Журналистика	4	Интерес выражен слабо
Юриспруденция	6	Выраженный интерес
Педагогика	2	Интерес выражен слабо
Сфера обслуживания, торговля	1	Интерес выражен слабо
Физкультура и спорт	2	Интерес выражен слабо
Музыка	3	Интерес выражен слабо
Сценическое искусство	2	Интерес выражен слабо
Изобразительное искусство	4	Интерес выражен слабо
Экология	4	Интерес выражен слабо

[Вернуться на главную страницу](#)

[Сохранить результат на эл. почту](#)

Рисунок 2.28 – Результат тестирования методики «Карта интересов»

Таким образом, был реализован личный кабинет респондента. С его помощью респондент может пройти тестирование по профориентационным методикам тестирования, которые созданы в системе и сохранить полученный результат на почту в виде письма.

2.5 Разработка и внедрение шаблона писем для респондента

Одним из требований заказчика является сохранение результата профориентационного тестирования респондента на почту. Необходимо реализовать данную функцию таким образом, чтобы письмо, получаемое пользователем, было выдержано в корпоративном стиле ХТИ – филиала СФУ, содержало в себе контактные данные необходимые потенциальному абитуриенту, а также было адаптивным, то есть хорошо отображаться на различных устройствах и почтовых клиентах. Кроме того, требуется добавить элементы персонализации в шаблон письма, такие как имя получателя. Это поможет сделать письмо более индивидуальным и привлекательным.

Первым этапом необходимо разработать структуру для шаблона. Это важно, так как грамотное разбиение контента позволяет создать логическую навигацию по письму. Хорошо структурированный контент помогает получателю легко перемещаться по письму, находить нужную информацию и быстро переходить к интересующим его разделам или действиям. Кроме того, это важно для адаптации письма. С учетом того, что многие пользователи просматривают электронную почту на мобильных устройствах, важно разбить контент таким образом, чтобы он был удобочитаемым на маленьких экранах. Это включает использование кратких абзацев, списков и более компактного формата для доставки информации. Несмотря на то, что процесс создания шаблона для письма на первый взгляд схож с процессом создания веб-страницы, данный процесс имеет свои специфические особенности и требует учёта факторов, связанных с ограничениями среды.

Так как каждая методика профориентационного тестирования имеет свой вариант вывода результата, необходимо разработать шаблон писем для каждой методики отдельно. Варианты писем приведены в приложении Б. На рисунке 2.29 представлено письмо, сформированное по шаблону письма.

Тема письма



proftest-khti@yandex.ru profitest-khti@yandex.ru Сегодня в 11:17
Я >

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ | SIBERIAN FEDERAL UNIVERSITY | ХАКАССКИЙ ТЕХНИЧЕСКИЙ ИНСТИТУТ филиал СФУ

Результаты теста «Методика Дж. Холланда»

Здравствуйте, Антон Филинов!

Все результаты

Артистический – сложный взгляд на жизнь, гибкость и независимость в принятии решений. Часто свойственен фатализм. Очень чувствителен, не социален, оригинален. Имеет богатое воображение, склонности к творческой деятельности, обладает хорошей интуицией, независим, эмоционален. Предпочитает занятия творческого характера. Преобладают вербальные способности. Для этого типа характерны исключительные способности восприятия и моторики, высокая чувствительность всех анализаторов. Имеет высокий жизненный идеал, нетривиален. Люди этого типа оригинальны, независимы в принятии решений, редко ориентируются на социальные нормы и одобрение, обладают необычным взглядом на жизнь, гибкостью мышления, эмоциональной чувствительностью. Отношения с людьми строят, опираясь на свои ощущения, эмоции, воображение, интуицию. Они не выносят жесткой регламентации, предпочитая свободный график работы. Часто выбирают профессии, связанные с литературой, театром, кино, музыкой, изобразительным искусством (писатель, фотограф, музыкант, художник, певец, журналист, архитектор, актер, дизайнер).

Близкие типы : интеллектуальный и социальный.

Противоположный тип : конвенциальный.

[Сайт ХТИ — филиала СФУ](#) | [Группа VK для поступающих](#) | [Написать нам на электронную почту](#)

Подробнее ознакомится с информацией о направлениях подготовки, начала приемной кампании, проходных баллах поступления и т.д. можно на [официальном сайте ХТИ – филиала СФУ](#) и группе [VK «Для поступающих ХТИ — филиал СФУ»](#)

Для личной консультации можно обратиться в приемную кампанию.
Адрес: 655017, Республика Хакасия, г. Абакан, ул. Щетинкина, 27 корпус А 9, ауд. 107А.

График работы приемной комиссии на площадке ХТИ – филиала СФУ:
Понедельник-пятница: с 08:00 до 17:00, обед с 12:00 до 13:00
Выходные: суббота, воскресенье.

Электронная почта: rk-khti@mail.ru
Приемная комиссия ХТИ – филиала СФУ: +7(3902)22-05-02, +7(983)054-41-72

[Отписаться от рассылки](#)

Рисунок 2.29 – Пример письма для сохранения результата

Далее представлен код программы, который отвечает за реализацию процесса отправки письма (рис. 2.30) на примере методики «Карта интересов».

Вначале функция получает пользователя и тест по их идентификаторам из базы данных. Также получается информация о отправителе письма из таблицы *Sender*. В роли отправителя выступает ХТИ – филиал СФУ.

Затем функция извлекает необходимые данные о сервере *SMTP* (*Simple Mail Transfer Protocol*) для отправки письма, такие как адрес сервера, порт, имя пользователя и пароль для аутентификации. Также получается адрес отправителя и адрес получателя из соответствующих полей.

```
@app.route("/send_email/<survey_id>/<user_id>")
def send_email(survey_id, user_id):
    # Получение пользователя по идентификатору (user_id) и опроса по идентификатору (survey_id)
    user = User.query.get(user_id)
    survey = Survey.query.get(survey_id)
    sender = Sender.query.get(1)

    smtp_server = sender.smtp_server
    smtp_port = sender.smtp_port
    smtp_username = sender.smtp_username
    smtp_password = sender.smtp_password
    sender_email = sender.sender_email
    receiver_email = user.email

    # Создаем объект MIMEMultipart для письма
    message = MIMEMultipart("alternative")
    message["Subject"] = "Тема письма"
    message["From"] = sender_email
    message["To"] = receiver_email

    if "Карта" in survey.name:
        data = custom_handlers.map_of_interests(survey_id, user_id)
        letter = render_template(
            "/email/map_of_interests.html",
            map_of_interests=data[0],
            best_interest=data[1],
        )

    letter = render_template(
        "/email/map_of_interests.html",
        map_of_interests=data[0],
        best_interest=data[1],
        user=user,
    )

    # Создаем объект MIMEText с HTML-контентом
    html_content = MIMEText(letter, "html")
    message.attach(html_content)

    # Устанавливаем соединение с SMTP-сервером и отправляем письмо
    with smtplib.SMTP(smtp_server, smtp_port) as server:
        # server.ehlo()
        server.starttls()
        server.login(smtp_username, smtp_password)
        server.sendmail(sender_email, receiver_email, message.as_string())

    return redirect("/")
```

Рисунок 2.30 – Отправка письма пользователю

Далее создается объект *MIMEMultipart*, который представляет собой контейнер для альтернативных версий письма и содержит метаданные письма, такие как тема, отправитель и получатель.

После создания объекта *MIMEMultipart* и формирования письма с альтернативными версиями контента, необходимо добавить *HTML*-контент к письму. Для этого создается объект *MIMEText* с *HTML*-контентом, где *letter* содержит сгенерированный ранее *HTML*-шаблон письма. Затем этот объект *MIMEText* с *HTML*-контентом прикрепляется к объекту *message* с помощью метода *attach*.

Далее следует блок кода, который устанавливает соединение с *SMTP*-сервером и отправляет письмо.

Таким образом, данный код устанавливает соединение с *SMTP*-сервером, аутентифицируется и отправляет письмо с заданным *HTML*-контентом получателю.

Вывод по разделу «Описание процесса разработки информационной системы профориентационного тестирования»

В ходе выполнения раздела выполнена настройка среды разработки, созданы база данных, форма регистрации и авторизации. Для каждого пользователя разработан личный кабинет в соответствии с его функциями работы с системой. Для разработки серверной части использовался фреймворк *Flask*, в роли СУБД выступала *SQLite*.

Для разработки клиентской части использовались языки *HTML*, *CSS* и *JavaScript*.

Была выполнена настройка клиент-серверного взаимодействия между обеими частями. Таким образом, был получен целостный программный продукт, функционал которого был протестирован от лица сотрудника ОДПиНН, *super*-пользователя и респондента.

3 Оценка экономической эффективности разработки и внедрения информационной системы

Для оценки экономической эффективности выбран метод «*Total Cost of Ownership*» (TCO), который учитывает общую стоимость владения информационной системой на протяжении всего её жизненного цикла. Для оценки затрат на реализацию проекта необходимо учесть капитальные, эксплуатационные и прямые затраты, а также определить экономическую эффективность проекта.

В рамках расчёта затрат на проект были определены сроки его выполнения, включающие консультации с заказчиком и анализ предметной области, а также разработку ИС.

Кроме того, в начале работы над проектом невозможно точно предвидеть, какие проблемы могут возникнуть в процессе его выполнения. Чтобы избежать потенциального ущерба, необходимо определить возможные риски проекта. Для этого требуется собрать и проанализировать информацию о возможных негативных исходах, чтобы выявить общие тенденции и закономерности их проявления.

3.1 Капитальные затраты

Капитальные затраты на ИС носят разовый характер и вычисляются по следующей формуле

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (1)$$

где $K_{\text{пр}}$ – затраты на проектирование ИС;

$K_{\text{тс}}$ – затраты на технические средства управления;

$K_{\text{лс}}$ – затраты на создание линий связи локальных сетей;

$K_{\text{по}}$ – затраты на программные средства;

$K_{\text{но}}$ – затраты на формирование информационной базы данных;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

3.1.1 Затраты на проектирование

Затраты на проектирование ИС рассчитываются по следующей формуле

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{ипс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (2)$$

где $K_{\text{зп}}$ – затраты на заработную плату проектировщиков;

$K_{\text{ипс}}$ – затраты на инструментальные программные средства;

$K_{\text{свт}}$ – затраты на средства вычислительной техники;

$K_{\text{проч}}$ – прочие затраты.

Расчет заработной платы. Для создания информационной системы требуется один программист, который разработает базу данных, дизайн интерфейса, логику работы ИС, а также выполнит администрирование прав пользователей ИС.

Для выплаты заработной платы программисту используется тарифный расчет оплаты труда, который соответствует установленной минимальной оплате труда (МРОТ) в Республике Хакасия [18]. Следовательно, заработная плата (ЗП) специалиста составляет 16 242 руб., не включая налог на доходы физических лиц (НДФЛ), северный коэффициент (30%) и районный коэффициент (30%).

Срок работы программиста составляет 1 месяц (22 рабочих дня).

Расчёт заработной платы программиста представлен в таблице 3.

Таблица 3 – Расчёт заработной платы

В рублях

Доходы		Расходы	
Оклад	16 242		
Северный коэффициент	4 873		
Районный коэффициент	4 873		
Итого месяц	25 987	НДФЛ	3 378
Итого в час	148		
Итого на руки	22 609		

$K_{зп}$ представляет собой сумму, которая включает в себя не только заработную плату, но также и дополнительные средства, необходимые для оплаты обязательных страховых взносов. Общий объем этих отчислений составляет 30,2% от заработной платы. Таким образом, $K_{зп}$ составляет 33 835 руб.

Затраты на программные средства. Далее необходимо рассчитать затраты на инструментальные программные средства для разработки информационной системы. В таблице 4 представлен перечень необходимого программного обеспечения (ПО).

Таблица 4 – Перечень программного обеспечения

Наименование ПО	Цена, руб.
Windows 11 Home	15999
Pixso	0
SQLite Studio	0
Visual Studio Code	0
Python	0
Браузер	0
Open Office	0

Сумма расходов на программное обеспечение будет включать только стоимость лицензионного пакета операционной системы Microsoft Windows 11 Home. В связи с тем, что разработка информационной системы займет 1 месяц, необходимо учесть амортизационную стоимость за период использования.

Для начала была рассчитана годовая норма амортизации по формуле (3).

$$H_a = \frac{1}{\text{СПИ}} \times 100\%, \quad (3)$$

где СПИ – срок полезного использования.

Для вычисления амортизации программного обеспечения за год используется формула (4).

$$A_{\text{год}} = C_t \times H_a, \quad (4)$$

где C_t – стоимость оборудования;

H_a – норма амортизации.

Для расчёта амортизации на проектирование используется формула (5).

$$A_{\text{пр}} = \frac{(A_{\text{год}} \times K_{\text{дэ}})}{12}, \quad (5)$$

где $K_{\text{дэ}}$ – количество месяцев эксплуатации.

Таким образом, $K_{\text{ипс}} = A_{\text{пр}} = 444$ руб.

Затраты на средства вычислительной техники. Для разработки информационной системы из оборудования использовался только ноутбук со следующими характеристиками:

- процессор: Ryzen 3 3200U;
- графический процессор: Radeon Vega 3 Graphics;
- частота процессора: 2.6 ГГц (3.5 ГГц, в режиме Turbo);
- объём оперативной памяти: 8 ГБ;
- объём встроенной памяти: 256 ГБ.

Стоимость данного ноутбука составляет 39 999 руб.

В силу того, что ноутбук представляет собой более сложное техническое устройство, вероятность возникновения поломок у него выше, поэтому минимальный срок его службы равен 3 годам.

Для определения ставки амортизации за год на вычислительную технику были использованы формулы (3) и (4).

Таким образом, $K_{\text{свт}} = 1111$ руб.

Расчёт прочих расходов. Для прочих расходов было выделено 3% от общей суммы проектных затрат. $K_{\text{проч}} = 1062$ руб.

Затраты на проектирование информационной системы профориентационного тестирования представлены в таблице 5.

Таблица 5 – Затраты на проектирование ИС

Наименование затрат	Затраты, руб.
Затраты на заработную плату	33 835
Затраты на инструментальные программные средства	444
Затраты на средства вычислительной техники	1 111
Прочие затраты	1 062
Итого	36 452

3.1.2 Расчёт капитальных затрат

Расчёт капитальных затрат будет выполнен по формуле (1), которая представлена в п. 3.1.

$K_{\text{пр}}$ подсчитаны в пункте 3.1.1 и составляют 36 452 руб.

$K_{\text{тс}}$ равны 0 руб., поскольку амортизационные расходы, связанные с ними, являются крайне незначительными и не учитываются.

$K_{\text{лс}}$ равны 0 руб., так как расходы по использованию услуг интернет-провайдера уже включены организацией в постоянные затраты, в ходе реализации иной работы.

$K_{но}$ равны 0 руб., так как для работы ИС для профориентационного тестирования необходимо бесплатное ПО, а именно веб-браузер для клиентской стороны и веб-сервер для серверной.

$K_{но}$ равны 0 руб., так как информационная база была разработана на этапе проектирования.

$K_{об}$ равны затратам на заработную плату программиста 1 рабочего дня, так как данного времени достаточно для того, чтобы объяснить сотрудникам ХТИ – филиала СФУ, которые будут работать с системой, основные принципы работы с ИС для профориентационного тестирования и ответить на все вопросы. $K_{об} = 1538$ руб.

$K_{ос}$ включают в себя заработную плату программиста за время опытной эксплуатации, которая составит 2 дня. Данное время необходимо, чтобы программист мог выполнить тестирование и исправить найденные недочёты. $K_{ос} = 3076$ руб.

Список капитальных затрат, а также итоговая сумма представлены в таблице 6.

Таблица 6 – Список капитальных затрат

Наименование затрат	Затраты, руб.
Проектные затраты	36 452
Затраты на технические средства для эксплуатации	0
Затраты на линии связи	0
Затраты на программное обеспечение	0
Затраты на формирование информационной базы	0
Затраты на обучение персонала	1 538
Затраты на опытную эксплуатацию	3 076
Итого	41 066

3.2 Расчёт эксплуатационных затрат

Эксплуатационные расходы, в отличие от капитальных расходов, возникают регулярно и рассчитываются на основе годовой суммы. Для определения этих расходов используется следующая формула

$$C = C_{\text{зп}} + C_{\text{ао}} + C_{\text{то}} + C_{\text{лс}} + C_{\text{ни}} + C_{\text{проч}}, \quad (6)$$

где $C_{\text{зп}}$ – зарплата персонала, работающего с ИС;

$C_{\text{ао}}$ – амортизационные отчисления;

$C_{\text{то}}$ – затраты на техническое обслуживание;

$C_{\text{лс}}$ – затраты на использование глобальных сетей;

$C_{\text{ни}}$ – затраты на носители информации;

$C_{\text{проч}}$ – прочие затраты.

Заработная плата персонала, работающего с ИС. С данной информационной системой будут работать два специалиста: младший программист и специалист по учебно-методической работе. Для входа в систему и открытия доступа к тестам младший программист будет тратить не больше 5 минут своего рабочего времени один-два раз в месяц, а также, примерно, 7 часов в месяц на обслуживание системы. Специалист по учебно-методической работе в зависимости от методики тестирования будет, примерно, тратить от 60 до 90 минут своего рабочего времени на проведение тестирования в рамках «Дня Открытых дверей» (среднее время прохождения теста 30 минут), а также 10 минут для выгрузки отчёта из системы один раз в месяц. Таким образом, младший программист будет тратить в год 1 рабочий день, а специалист по учебно-методической работе 2 рабочих дня в год. Исходя из времени работы был выполнен расчёт заработной платы каждого специалиста.

Таблица 7 – Заработная плата персонала

Должность	Оклад	Районный коэффициент	Северный коэффициент	НДФЛ	Итого в месяц	Итого за день	Итого (без учёта НДФЛ)
Младший программист	20390	6117	6117	4241	32624	1483	28383
Специалист по учебно-методической работе	16310	4893	4893	3393	26096	1186	22704

С учётом отчислений на страховые взносы показатель $C_{зп} = 5020$ руб.

Затраты на амортизационные отчисления. Поскольку система будет функционировать на сервере института, а для работы с ней специалисты будут использовать компьютеры, которые они используют для своей основной работы, расходы на амортизацию не потребуются: $C_{ао} = 0$ руб.

Затраты на техническое обслуживание. Поскольку информационная система может требовать обслуживания, за которое будет отвечать младший программист, необходимо запланировать 1 рабочий день в год на эти цели. Таким образом, $C_{то} = 1931$ руб.

Затраты на использование глобальных сетей. ХТИ – филиал СФУ уже подключен к глобальным сетям, поэтому $C_{лс} = 0$ руб.

Затраты на носители информации. В проекте не требуется хранить огромные объемы данных в базе данных, поэтому для её хранения достаточно использовать накопитель на сервере института: $C_{ни} = 0$ руб.

Прочие затраты. Прочие затраты обычно составляют от 3 до 5 процентов от общей суммы эксплуатационных затрат. В случае данного проекта оптимальным значением будет 3%: $C_{проч} = 209$ руб.

Список эксплуатационных затрат и итоговая сумма представлены в таблице 8.

Таблица 8 – Эксплуатационные затраты

Наименование затрат	Затраты, руб.
Зарплата персонала, работающего с ИС	5 020
Амортизационные отчисления	0
Затраты на техническое обслуживание	1 931
Затраты на использование глобальных сетей	0
Затраты на носители информации	0
Прочие затраты	209
Итого	7160

3.3 Совокупная стоимость владения системой

Методика *TCO* (*Total Cost of Ownership* – совокупная стоимость владения) была создана компанией *Gartner Group* в конце 1980-х годов для определения финансовых затрат на обладание продуктом *IT*. Для расчета совокупной стоимости владения информационной системой используется следующая формула

$$TCO = DE + IC_1 + IC_2, \quad (7)$$

где *DE* (*direct expenses*) – прямые затраты;

IC₁, IC₂ (*indirect costs*) – косвенные затраты первой и второй группы.

Для данного проекта косвенные затраты не являются значимыми, следовательно $TCO \approx DE$. Прямые затраты рассчитываются по следующей формуле

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (8)$$

где *DE₁* – капитальные затраты;

DE₂ – расходы на управление ИТ;

DE₃ – расходы на техническую поддержку АО и ПО;

DE₄ – расходы на разработку прикладного ПО внутренними силами;

DE₅ – расходы на аутсорсинг;

DE_6 – командировочные расходы;

DE_7 – расходы на услуги связи;

DE_8 – другие группы расходов.

Проанализировав все данные, были выявлены следующие затраты

$$DE_1 = K = 41066 \text{ руб.};$$

$$DE_2 = C_{\text{сп}} = 5020 \text{ руб.};$$

$$DE_3 = C_{\text{то}} + C_{\text{ао}} = 1931 \text{ руб.};$$

$$DE_4 = 0 \text{ руб.};$$

$$DE_5 = 0 \text{ руб.};$$

$$DE_6 = 0 \text{ руб.};$$

$$DE_7 = C_{\text{лс}} = 0;$$

$$DE_8 = C_{\text{проч}} + C_{\text{ни}} = 209 \text{ руб.}$$

Таким образом, исходя из данных, представленных выше, $ТСО \approx 48226$ руб.

3.4 Оценка рисков реализации проекта

Для предотвращения возможных проблем в процессе реализации проекта используется практика оценки рисков. Эта методика включает анализ потенциальных проблем, их вероятности возникновения, уровня влияния на проект, а также разработку стратегий решения. При грамотной оценке рисков возможные проблемы не будут неожиданными, так как будут заранее продуманы пути их решения.

1. Вероятность возникновения риска соответствия низкая, так как в начальной стадии проекта было проведено моделирование ИС, в рамках которого был разработан необходимый функционал и пользовательские сценарии

взаимодействия. Тем не менее, необходимо учитывать возможность недопонимания с заказчиком, что может повлиять на риск соответствия.

2. Риск задержки сроков реализации проекта имеет средний уровень вероятности. Разработчик проекта оценивает сложность реализации и необходимый срок на этапе планирования. Тем не менее, разработка информационной системы – это сложный и многогранный процесс, который может столкнуться с неожиданными проблемами, поэтому необходимо предусмотреть дополнительное время на их решение.

3. Вероятность возникновения риска эксплуатации низкая, так как сотрудники проводят необходимые технические работы по мере необходимости.

4. Вероятность возникновения риска потери данных программного продукта средняя, так как разработчик делает резервные копии на протяжении всего процесса разработки, поэтому имеет возможность восстановить проект в случае необходимости.

В таблице 9 приведен анализ потенциальных рисков проекта.

Таблица 9 – Перечень рисков

Группа рисков	Перечень рисков проекта	Уровень влияния	Вероятность	Возможные варианты решения
Риск соответствия	Недостаточное взаимопонимание между заказчиком и разработчиком	Высокая	Низкая	Консультация разработчика с заказчиком
Риск задержки сроков реализации проекта	Непредвиденные расходы, срыв сроков внедрения	Средний	Средняя	Заложить дополнительное время на решение непредвиденных технических проблем
Риск эксплуатации	Непредвиденный сбой работы сервера	Высокий	Низкий	Проведение технических работ по обслуживанию сервера
Риск потери данных проекта	Непредвиденная поломка компьютера, на котором выполняется процесс разработки	Высокий	Средняя	Подготовить среду разработки для резервного копирования проекта

3.5 Экономическая эффективность реализации проекта

Данный проект не предполагает получения прибыли и является бесплатным. Его основная цель – упростить рабочий процесс сотрудника ОДПиНН через автоматизацию процессов. Реализация проекта способствует повышению эффективности работы сотрудников и может привести к уменьшению их нагрузки. Однако, автоматизация процессов освобождает сотрудников от рутинной работы, такой как ручной анализ результатов тестирования и составление отчетов, что позволяет им сосредоточиться на более важных задачах, соответствующих их должностным обязанностям. Таким образом, сэкономленные средства могут быть использованы ХТИ – филиалом СФУ для дальнейшего распределения.

В качестве программы для сравнения при разработке проекта приняты программы: «Отбор в профильные классы «ПрофильКЛАСС». Методика Г.В. Резапкиной» и «*Online Test Pad*». Были учтены следующие факторы: смежный профиль и соответствие требованиям технического задания проекта.

Сравнительный анализ представлен в таблице 10.

Таблица 10 – Сравнительный анализ проектов

Информационная система «ПрофТест ХТИ»	Отбор в профильные классы «Профиль-КЛАСС». Методика Г.В. Резапкиной	Online Test Pad
Интерфейс рассчитан на трёх пользователей: super-пользователь, специалист, респондент. Имеется возможность прохождения нескольких тестов в системе, а также при необходимости super-пользователь может создать в системе новый тест.	В системе предусмотрен интерфейс для двух типов пользователей: специалиста и респондента. Однако, есть ограничение интерфейса: специалист не обладает возможностью создания новых тестов в системе.	Предоставляется удобный интерфейс для создания новых тестов. Однако, недостатком данного интерфейса является то, что каждый тест открывается по отдельной ссылке. Требуется место для хранения ссылок, чтобы респонденты могли самостоятельно выбирать тесты.
Поддерживается ПК и мобильными устройствами	Поддерживается ПК	Поддерживается ПК и мобильными устройствами

Окончание таблицы 10

4 (имеется возможность обновления и добавления методик)	9	4 (имеется возможность обновления и добавления методик)
Для хранения данных используется локальная база данных	Для хранения данных используется локальная база данных	Система не предоставляет гарантий сохранения данных
Доступ осуществляется из сети Интернет	Доступ только на локальном ПК	Доступ осуществляется из сети Интернет
Программист обладает возможностью настраивать интерфейс системы в соответствии с корпоративным стилем института, включая изменение цветовой гаммы и добавление или изменение информации.	Нет возможности редактирования интерфейса.	Нет возможности редактирования интерфейса.
Интерфейс и функционал системы оптимизированы таким образом, чтобы не перегружать пользователя, что обеспечивает легкость в его использовании. Если возникнет необходимость, пользователь может обратиться за технической поддержкой к программисту.	В составе предлагаемого продукта включено руководство пользователя, которое поможет освоить его функциональность. Кроме того, для специалистов доступно платное обучение, позволяющее более глубоко овладеть системой.	На веб-сайте доступно пошаговое руководство по созданию теста. Если у вас возникнут вопросы, вы можете обратиться в службу технической поддержки для получения помощи.
Капитальные затраты	20 800 руб.	Бесплатно

Анализируя данные таблицы, можно сделать вывод, что разработка собственной системы имеет ряд преимуществ: функционал ИС соответствует требованиям заказчика, а также интерфейс системы выдержан в корпоративном стиле. Основным недостатком системы «Отбор в профильные классы «Профиль КЛАСС». Методика Г.В. Резапкиной» является локальное использование системы, что не удовлетворяет основному требованию заказчика. Использование информационной системы «*Online Test Pad*» невозможно по причине того, данная система не зарегистрирована официально в «Едином реестре российских программ для ЭВМ и БД».

Вывод по разделу «Оценка экономической эффективности разработки и внедрения информационной системы»

В данном разделе были выполнены расчеты затрат и проведена оценка рисков, связанных с реализацией проекта создания системы для профориентационного тестирования. Суммарные капитальные затраты на разработку ИС составили 41 066 рублей, что соответствует средней стоимости проектов данного уровня. Эксплуатационные расходы на использование ИС оценены в размере 7 160 рублей, что является хорошим показателем для данного вида затрат. Суммарная стоимость владения ИС для заказчика, начиная с момента начала разработки и заканчивая моментом внедрения, составит 48 226 рублей. Кроме того, был произведен анализ возможных рисков, связанных с реализацией проекта, а также рассмотрены способы уменьшения вероятности их возникновения. Система для профориентационного тестирования будет разработана для отдела довузовской подготовки и нового набора ХТИ - филиала СФУ согласно техническому заданию. Программный продукт не будет коммерчески распространяться, поскольку является заказным. Разработка данной системы улучшит эффективность работы сотрудников, что подтверждается коэффициентом экономической эффективности проекта.

ЗАКЛЮЧЕНИЕ

В рамках бакалаврской работы была разработана информационная система профориентационного тестирования для ХТИ – филиала СФУ. Система призвана помочь автоматизировать процесс проведения профориентационного тестирования в рамках мероприятий, организуемых ОДПиНН среди большого количества респондентов, что позволяет упростить работу сотрудников.

Кроме того, было выполнено проектирование бизнес-процесса для информационной системы, с помощью которого были определены основные роли пользователей системы и их функционал. Также был выполнен анализ средств разработки информационной системы и обоснован выбор программных средств: *SQLite* в качестве базы данных, *HTML*, *CSS* и *JavaScript* для разработки клиентской стороны и *Flask* – серверной стороны.

Разработан программный продукт, соответствующий требованиям заказчика:

- возможность добавления новых тестов на основе профориентационных методик, а также создание опросов для обратной связи с пользователями;
- открытие и закрытие доступа к существующим тестам в системе, а также удаление их из системы;
- редактирование существующих вопросов и удаление их из системы;
- просмотр результата тестирования пользователя по методике тестирования, а также просмотр общей статистики по результатам тестов в виде диаграмм;
- сохранение результата тестирования на почту респондента с использованием шаблона письма.

Выполнен расчёт экономической эффективности проекта и обоснована необходимость разработки информационной системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Хакасский технический институт – филиал Сибирского федерального университета : официальный сайт. – Абакан, 2023 – .URL: <https://khti.ru/institute/> (дата обращения: 10.02.2023).
2. ПСП ЦПЮИ – 2014 : Положение о центре подготовки юного инженера Хакасского технического института – филиала ФГАОУ ВПО «Сибирский федеральный университет» : официальный сайт. – Абакан, 2023. – URL: <https://khti.ru/documents/Институт/Положение%20о%20ЦПЮИ.pdf> (дата обращения: 10.02.2023).
3. ПСП ОДПиНН – 2014 : Положение об отделе довузовской подготовки и нового набора: сайт / Хакасский технический институт – филиала Сибирского федерального университета. – официальный сайт. – Абакан, 2023. – URL: <https://khti.ru/documents/Институт/Положение%20об%20ОДПиНН.pdf> (дата обращения: 10.02.2023).
4. Автоматизированная система по проведению профориентационного тестирования в образовательных организациях / Е.В. Кусакина, Е.А. Самарина, Ю.В. Фролов // Вестник РУДН. Серия: Информатизации образования. – 2019. – Т.16, №3. – С. 243-256.
5. Профориентация.ру // Что такое профориентация : [сайт]. – 2022. – URL: <https://proforientatsia.ru/> (дата обращения: 12.02.2023).
6. Навигатор образования // Состояния и перспективы системы профориентации : [сайт]. – 2023. – URL: https://fulledu.ru/articles/1407_sostoyanie-perspektivy-sistemy-proforientacii.html (дата обращения: 12.03.2023).
7. Соломин, И. Л. Современные методы психологической экспресс-диагностики и профессионального консультирования // СБП.: Речь. – 2006. – Т. 280. – С.8.
8. Резапкина, Г. В. Отбор в профильные классы / Г.В. Резапкина. – Издательский дом «Генезис», 2005. – 1124 с.
9. Методика «Карта интересов»: сайт / Государственное автономное

профессиональное образовательное учреждение Свердловской области «Ирбитский гуманитарный колледж». – Ирбит : ИГК, 2012 – URL: <https://irbitgc.ru/upload/files/section/project/ACC/KartaIntersov.pdf> (дата обращения: 14.03.2023).

10. Кетько, С. М., Пакулина, С. А. Методика диагностики мотивации учения студентов педагогического вуза // Психолого-педагогические исследования. – 2010. – Т. 2010. – № 1.

11. Тестометрика // Онлайн тесты: психологические, образовательные, IQ : [сайт]. – 2022. – URL: <https://testometrika.com/> (дата обращения: 15.04.2023).

12. Работа России // Профорентация : [сайт]. – 2023. – URL: <https://trudvsem.ru/proforientation> (дата обращения: 15.04.2023).

13. HELPIK.ORG : Методы моделирования бизнес процессов : официальный сайт. – 2023. – URL: <https://helpiks.org/7-74559.html> (дата обращения: 16.04.2023).

14. Веб-приложение // Википедия – свободная энциклопедия : [сайт]. – 2001. – URL: <https://ru.wikipedia.org/wiki/Веб-приложение> (дата обращения: 20.04.2023).

15. Django : официальный сайт. – 2023. – URL: <https://www.djangoproject.com/> (дата обращения: 21.04.2023).

16. Flask: официальный сайт. – 2023. – URL: <https://flask.palletsprojects.com/> (дата обращения: 21.04.2023).

17. Google for Developers // Charts : официальный сайт. – 2023. – URL: <https://developers.google.com/chart?hl=en> (дата обращения: 12.05.2023).

18. МРОТ в 2023 году по регионам России // Бухгалтерский интернет-журнал : [сайт]. – 2023. – URL: <https://buhguru.com/buhgalteria/zarplata/novyj-mrot-2023-tablicza-po-regionam.html> (дата обращения: 20.05.2023).

ПРИЛОЖЕНИЕ А

Логика вычисления результата для методик тестирования

```
from data_models import User, Question, Survey, Response

def map_of_interests(survey_id, user_id):
    user_responses = Response.query.filter_by(u_id=user_id,
s_id=survey_id).all()

    if len(user_responses) == 0:
        return 1 # "Для данного опроса и пользователя результатов не найде-
но."
    responses_data = []
    for response in user_responses:
        responses_data.append(response.num)
    interests = [
        "Биология",
        "Физика",
        "Химия",
        "География",
        "Медицина",
        "Техника и электроника",
        "Строительство",
        "Математика",
        "Экономика",
        "Иностранные языки",
        "Транспорт",
        "Авиация, Морское дело",
        "Военные специальности",
        "История",
        "Рабочие специальности",
        "Журналистика",
        "Юриспруденция",
        "Педагогика",
        "Сфера обслуживания, торговля",
        "Физкультура и спорт",
        "Музыка",
        "Сценическое искусство",
        "Изобразительное искусство",
        "Экология",
    ]
    # Исходный массив из 144 значений
    array = responses_data # Здесь должен быть ваш массив из 144 значений
```

```

# Разделение массива на 6 подмассивов по 24 значения
sub_arrays = [array[i : i + 24] for i in range(0, len(array), 24)]

# Преобразование значений в каждом подмассиве
for sub_array in sub_arrays:
    for i in range(len(sub_array)):
        if sub_array[i] == 1:
            sub_array[i] = 2
        elif sub_array[i] == 2:
            sub_array[i] = 1
        elif sub_array[i] == 3:
            sub_array[i] = 0
        elif sub_array[i] == 4:
            sub_array[i] = -1
        elif sub_array[i] == 5:
            sub_array[i] = -2

# Вычисление значений в каждом столбце и подготовка результата
map_of_interests = []
best_interest = {"interest": "Экономика", "score": -12}
for i in range(24):
    column_sum = sum(sub_array[i] for sub_array in sub_arrays)
    entry_of_interests = {"interest": interests[i], "score": column_sum}
    map_of_interests.append(entry_of_interests)
    if column_sum > best_interest["score"]:
        best_interest["interest"] = interests[i]
        best_interest["score"] = column_sum

result = [map_of_interests, best_interest]

return result

def form_for_orientation(survey_id, user_id):
    # Получение пользователя по идентификатору (user_id) и опроса по идентификатору (survey_id)
    user = User.query.get(user_id)
    survey = Survey.query.get(survey_id)

    # Получение всех ответов на вопросы для данного пользователя и опроса
    responses = Response.query.filter_by(u_id=user.id, s_id=survey.id).all()

    if len(responses) == 0:
        return "Для данного опроса и пользователя результатов не найдено."

    # Создание списка для хранения результатов ответов пользователя
    response_scores = []

```

```

# Проход по каждому ответу и добавление его результата в список
response_scores
for response in responses:
    response_scores.append(response.text or response.num)

# Алгоритм обработки результатов теста
want_category_scores = [0] * 7 # Оценки для каждой подкатегории в "Я
хочу"
can_category_scores = [0] * 7 # Оценки для каждой подкатегории в "Я мо-
гу"
categories = [
    "Человек-человек",
    "Человек-техника",
    "Человек-знаковая система",
    "Человек-художественный образ",
    "Человек-природа",
    "исполнительский",
    "творческий",
]

for index, response in enumerate(response_scores):
    if index < 35: # "Я хочу"
        subcategory_index = (
            index // 5
        ) # Индекс подкатегории, определяется делением на 5
        want_category_scores[subcategory_index] += response - 1
    else: # "Я могу"
        subcategory_index = (
            index - 35
        ) // 5 # Индекс подкатегории, определяется делением на 5 с сме-
щением на 35
        can_category_scores[subcategory_index] += response - 1

want_category = []
can_category = []

for i in range(7):
    temp = {}
    temp["name"] = categories[i]
    temp["score"] = want_category_scores[i]
    want_category.append(temp)

for i in range(7):
    temp = {}
    temp["name"] = categories[i]
    temp["score"] = can_category_scores[i]
    can_category.append(temp)

```

```

    want_result = [
        {
            "type": categories[
                want_category_scores[:5].index(max(want_category_scores[:5]))
            ]
        },
        {
            "mood": categories[
                want_category_scores[5:7].index(max(want_category_scores[5:7
])) + 5
            ]
        },
    ]

    can_result = [
        {
            "type": categories[
                can_category_scores[:5].index(max(can_category_scores[:5]))
            ]
        },
        {
            "mood": categories[
                can_category_scores[5:7].index(max(can_category_scores[5:7]
)) + 5
            ]
        },
    ]

    result = [want_category, want_result, can_category, can_result]
    return result

def holland_test(survey_id, user_id):
    user = User.query.get(user_id)
    survey = Survey.query.get(survey_id)

    # Получение всех ответов на вопросы для данного пользователя и опроса
    responses = Response.query.filter_by(u_id=user.id, s_id=survey.id).all()

    if len(responses) == 0:
        return "Для данного опроса и пользователя результатов не найдено."

    # Создание списка для хранения результатов ответов пользователя
    response_scores = []

    # Проход по каждому ответу и добавление его результата в список
    response_scores
    for response in responses:

```



```

        response_scores.append(response.num)

# Переменные для результатов по категориям
realist_score = 0
intelli_score = 0
social_score = 0
convent_score = 0
enterpr_score = 0
artist_score = 0

# Обработка результатов
for i in range(len(response_scores)):
    if response_scores[i] == 1:
        if i + 1 in [1, 2, 3, 4, 5, 16, 17, 19, 21, 28, 31, 32, 33, 34]:
            realist_score += 1
        elif i + 1 in [6, 7, 8, 9, 20, 22, 23, 24, 35, 36, 37]:
            intelli_score += 1
        elif i + 1 in [10, 11, 12, 25, 26, 27, 38, 39]:
            social_score += 1
        elif i + 1 in [13, 14, 18, 29, 40, 42]:
            convent_score += 1
        elif i + 1 in [15, 30]:
            enterpr_score += 1
        elif i + 1 in [41]:
            artist_score += 1
    elif response_scores[i] == 2:
        if i + 1 in [1, 16, 31]:
            intelli_score += 1
        elif i + 1 in [2, 6, 17, 20, 36, 41]:
            social_score += 1
        elif i + 1 in [3, 7, 10, 19, 22, 32, 35, 38]:
            convent_score += 1
        elif i + 1 in [4, 8, 11, 13, 18, 23, 25, 26, 28, 33, 39, 40]:
            enterpr_score += 1
        elif i + 1 in [5, 9, 12, 14, 15, 21, 24, 27, 29, 30, 34, 37,
42]:
            artist_score += 1

# Подготовка результатов
scores = [
    realist_score,
    intelli_score,
    social_score,
    convent_score,
    enterpr_score,
    artist_score,
]
names = [

```

```

        "реалистический",
        "интеллектуальный",
        "социальный",
        "конвенциональный",
        "предприимчивый",
        "артистический",
    ]
    all_scores = {names[i]: scores[i] for i in range(len(scores))}
    highest_score = max(
        [
            realist_score,
            intelli_score,
            social_score,
            convent_score,
            enterpr_score,
            artist_score,
        ]
    )

    result = [all_scores, highest_score]

    # Возвращение результата для шаблона
    return result

def matrix_of_professions(survey_id, user_id):
    user = User.query.get(user_id)
    survey = Survey.query.get(survey_id)

    # Получение всех ответов на вопросы для данного пользователя и опроса
    responses = Response.query.filter_by(u_id=user.id, s_id=survey.id).all()

    if len(responses) == 0:
        return "Для данного опроса и пользователя результатов не найдено."

    # Создание списка для хранения результатов ответов пользователя
    response_scores = []

    # Проход по каждому ответу и добавление его результата в список
    response_scores
    for response in responses:
        response_scores.append(response.num)

    sphere_num = []
    spheres = [
        "человек",
        "информация",
        "финансы",
        "техника",

```

```

        "искусство",
        "животные",
        "растения",
        "продукты",
        "изделия",
        "природные",
    ]
    kind_num = []
    kinds = [
        "управление",
        "обслуживание",
        "образование",
        "оздоровление",
        "творчество",
        "производство",
        "конструирование",
        "исследование",
        "защита",
        "контроль",
    ]

    for answer in response_scores[:10]:
        if answer == 2:
            sphere_num.append(0)
        elif answer == 1:
            sphere_num.append(1)

    for answer in response_scores[10:20]:
        if answer == 2:
            kind_num.append(0)
        elif answer == 1:
            kind_num.append(1)

    answers = []

    for i in range(10):
        if sphere_num[i]:
            for j in range(10):
                if kind_num[j]:
                    temp_answer = {}
                    temp_answer["sphere"] = spheres[i]
                    temp_answer["kind"] = kinds[j]
                    answers.append(temp_answer)


    return answers


```

ПРИЛОЖЕНИЕ Б

Варианты писем с результатами профориентационного тестирования


Тема письма

PR proftest-khti@yandex.ru proftest-khti@yandex.ru  Сегодня в 11:13
Я >



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

SIBERIAN
FEDERAL
UNIVERSITY



ХАКАССКИЙ
ТЕХНИЧЕСКИЙ
ИНСТИТУТ
филиал СФУ

Результаты теста «Анкета ориентации»

Здравствуйте, Антон Филинов!

Все результаты

Больше всего вам подходит исполнительский тип профессий ("Я хочу", Человек-знаковая система)

Среди них встречаются:

Дозировщик, Изготовитель трафаретов и плат, Кассир, Машинистка, Наборщик, Печатник, Телефонист

Больше всего вам подходит творческий тип профессий ("Я могу", Человек-человек)

Среди них встречаются:

Врач, Инспектор по кадрам, Менеджер, Методист по дошкольному воспитанию
Преподаватель, Психолог, Референт, Тренер-преподаватель, Учитель, Юрисконсульт

[Сайт ХТИ — филиала СФУ](#) | [Группа VK для поступающих](#) | [Написать нам на электронную почту](#)

Подробнее ознакомится с информацией о направлениях подготовки, начала приёмной кампании, проходных баллах поступления и т.д. можно на [официальном сайте ХТИ – филиала СФУ](#) и группе [VK «Для поступающих ХТИ — филиал СФУ»](#)


Для личной консультации можно обратиться в приёмную кампанию.
Адрес: 655017, Республика Хакасия, г. Абакан, ул. Цетинкина, 27 корпус А , ауд. 107А.

График работы приемной комиссии на площадке ХТИ – филиала СФУ:
Понедельник-пятница: с 08:00 до 17:00, обед с 12:00 до 13:00
Выходные: суббота, воскресенье.

Электронная почта: pk-khti@mail.ru
Приемная комиссия ХТИ – филиала СФУ: [+7\(3902\)22-05-02](tel:+73902220502), [+7\(983\)054-41-72](tel:+79830544172)

[Отписаться от рассылки](#)

Рисунок Б.1 – Письмо с результатом тестирования по методике «Анкета ориентация»

Тема письма



proftest-khti@yandex.ru prof-test-khti@yandex.ru Сегодня в 11:12
Я >



СИБИРСКИЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ | SIBERIAN
FEDERAL
UNIVERSITY



Результаты теста «Матрица профессий»

Здравствуйте, Антон Филинов!

Все результаты

Вам могут подойти следующие категории профессий:

Человек—Управление

Менеджер по персоналу, Администратор

Человек—Обслуживание

Продавец, Парикмахер, Официант

Человек—Творчество

Режиссёр, Артист, Музыкант

Информация—Управление

Маркетолог, Диспетчер, Статистик

Информация—Обслуживание

Переводчик, Экскурсовод, Библиотекарь

Информация—Творчество

Программист, Редактор, Web-дизайнер

Искусство—Управление

Режиссёр, Продюсер, Дирижёр

Искусство—Обслуживание

Гримёр, Костюмер, Парикмахер

Искусство—Творчество

Художник, Писатель, Композитор


[Сайт ХТИ — филиала СФУ](#) | [Группа VK для поступающих](#) | [Написать нам на электронную почту](#)

Подробнее ознакомьтесь с информацией о направлениях подготовки, начала приёмной кампании, проходных баллах поступления и т.д. можно на [официальном сайте ХТИ – филиала СФУ](#) и группе VK [«Для поступающих ХТИ — филиал СФУ»](#)

Для личной консультации можно обратиться в приёмную кампанию.

Адрес: 655017, Республика Хакасия, г. Абакан, ул. Щетинкина, 27 корпус А 9, ауд. 107А.

Рисунок Б.2 – Письмо с результатом тестирования по методике
«Матрица профессий»



СИБ ИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ | **SIBERIAN FEDERAL UNIVERSITY** | **ХАКАССКИЙ ТЕХНИЧЕСКИЙ ИНСТИТУТ** филиал СФУ

Результаты теста «Карта интересов»

Здравствуйете, Антон Филинов!

Больше всего вам подходит: История со счётом 10 очков.

Все результаты

Биология	3
Физика	0
Химия	-5
География	-4
Медицина	8
Техника и электроника	3
Строительство	-1
Математика	2
Экономика	3
Иностранные языки	2
Транспорт	5
Авиация, Морское дело	5
Военные специальности	2
История	10
Рабочие специальности	7
Журналистика	4
Юриспруденция	6
Педагогика	2
Сфера обслуживания, торговля	1
Физкультура и спорт	2
Музыка	3
Сценическое искусство	2
Изобразительное искусство	4
Экология	4

[Сайт ХТИ — филиала СФУ](#) | [Группа VK для поступающих](#) | [Написать нам на электронную почту](#)

Подробнее ознакомится с информацией о направлениях подготовки, начала приёмной кампании, проходных баллах поступления и т.д. можно на [официальном сайте ХТИ — филиала СФУ](#) и группе VK [«Для поступающих. ХТИ — филиал СФУ»](#)

Для личной консультации можно обратиться в приёмную кампанию.
Адрес: 655017, Республика Хакасия, г. Абакан, ул. Щетинкина, 27 корпус А 📍, ауд. 107А.

График работы приемной комиссии на площадке ХТИ – филиала СФУ:
Понедельник-пятница: с 08:00 до 17:00, обед с 12:00 до 13:00
Выходные: суббота, воскресенье.

Электронная почта: pk-khti@mail.ru
Приемная комиссия ХТИ – филиала СФУ: [+7\(3902\)22-05-02](tel:+73902220502), [+7\(983\)054-41-72](tel:+79830544172)

Рисунок Б.3 – Письмо с результатом тестирования по методике «Карта интересов»

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 18 наименований.

Один экземпляр сдан на кафедру.

« ____ » _____ 2023 г.

_____ Доронина Олеся Анатольевна
подпись

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, естественно-научных
и гуманитарных дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Папина

подпись

« 19 » июня 2023 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка информационной системы профориентационного тестирования
для ХТИ – филиала СФУ

Руководитель М. А. Буреева 19.06.23г. доцент, канд. физ-мат. наук М. А. Буреева
подпись, дата

Выпускник О. А. Доронина 19.06.2023
подпись, дата

Консультанты
по разделам:

Экономический М. А. Буреева 19.06.23г.
подпись, дата

Нормоконтролер А. Н. Кадычегова 19.06.23
подпись, дата

Абакан 2023