

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им Н.Э. Баумана)

Московский техникум космического приборостроения

Отделение Информационные системы и программирование

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

по профессиональному модулю ПМ.01 Разработка модулей программного обеспечения для компьютерных систем в объеме 108 часов

Код, Специальность 09.02.07 Информационные системы и программирование квалификация - Программист

Место прохождения практики ПАО «Ил»
(полное название организации)

Выполнил студент Дюрягина Ирина Валерьевна
(фамилия, имя, отчество)

Курс 4 Группа ТИП-82

Подпись студента

Оценка отлично

Дата приема отчета 10 апреля 2025г.

Руководитель практики от техникума
(подпись)

(фамилия, имя, отчество)

2025 г.

Московский техникум космического приборостроения МГТУ имени Н.Э. Баумана

ЗАДАНИЕ **на прохождение производственной практики**

на предприятии ПАО «Ил»

по: ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Студент Дюрягина Ирина Валерьевна 09.02.07 Информационные системы и
программирование квалификация - Программист ТИП-82
(фамилия, имя, отчество; индекс специальности, группа)

Студент во время прохождения учебной практики с «24» марта 2025 года по «12» апреля 2025 года

должен:

Ознакомиться:

1. С основными этапами разработки программного обеспечения.
2. С основными принципами технологии структурного и объектно-ориентированного программирования.
3. Со способами оптимизации и приемами рефакторинга.
4. С основными принципами отладки и тестирования программных продуктов.

Уметь:

1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней.
2. Создавать программу по разработанному алгоритму как отдельный модуль.
3. Выполнять отладку и тестирование программы на уровне модуля.
4. Осуществлять разработку кода программного модуля на современных языках программирования.
5. Выполнять оптимизацию и рефакторинг программного кода.
6. Оформлять документацию на программные средства языков программирования.

Получить практический опыт:

1. В разработке кода программного продукта на основе готовой спецификации на уровне модуля.

2. В использовании инструментальных средств на этапе отладки программного продукта.
3. В проведении тестирования программного модуля по определенному сценарию.
4. В разработке мобильных приложений.

По итогам производственной практики студент обязан представить:

1. Отчет по производственной практике (Титульный лист, Задание, отчет, Аттестационный лист, Дневник по практике).

Дата выдачи задания «12» марта 2025 г.

Руководитель практики от техникума

Гру Тамара О.В.

Студент

СРБ 10.04.25
(подпись, дата)

Дюрягина И. В.
(Фамилия И.О.)

СОДЕРЖАНИЕ

1. Теоретическое обоснование разработки программного продукта	5
1.1 Моделирование проектируемой системы	5
1.2 Разработка функциональных требований к системе.....	6
1.2.1 Функциональное назначение	6
1.2.2 Эксплуатационное назначение	7
1.2.3 Требования к функциональным характеристикам	8
1.2.4 Входные и выходные данные.....	10
1.2.5 Требования к надежности	11
1.2.6 Требования к техническим средствам	12
1.2.7 Требования к программной совместимости	13
1.2.8 Техническая документация	14
1.2.9 Пользовательская документация	15
1.2.10 Эксплуатационная документация.....	16
1.3 Обоснование выбора средств реализации программной системы	17
1.3.1 Анализ требований к клиентской части.....	17
1.3.2 Серверная архитектура системы.....	18
1.3.3 Система хранения данных.....	19
1.3.4 Взаимодействие компонентов системы	20
1.3.5 Дополнительные технологии	21
2. Разработка программного продукта.....	22
2.1 Разработка структуры данных.....	22
2.1.1 Модель данных и принципы проектирования	22
2.1.2 Детальное описание сущностей.....	24
2.2 Схема алгоритма основной программы	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27

1. Теоретическое обоснование разработки программного продукта

1.1 Моделирование проектируемой системы

На рисунке 1.1 представлена контекстная диаграмма IDEF0, показывающая главную задачу, которую решает выполнение бизнеспроцесса.

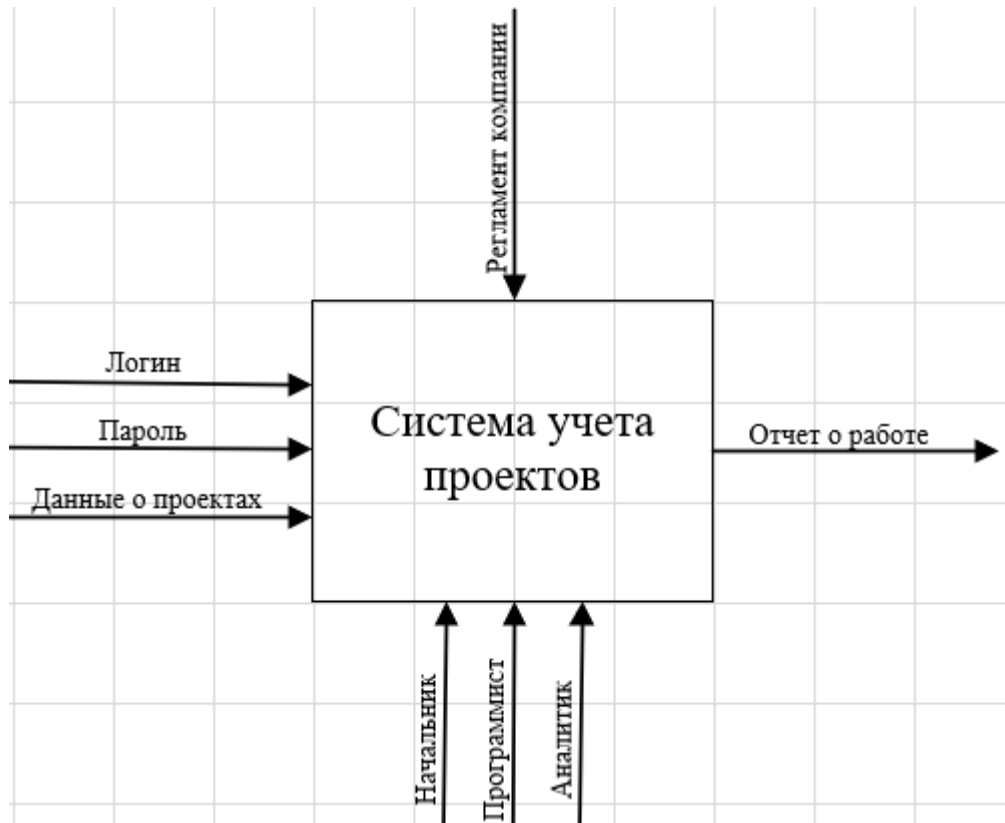


Рисунок 1.1 – Контекстная диаграмма IDEF0

1.2 Разработка функциональных требований к системе

1.2.1 Функциональное назначение

Разрабатываемая программная система представляет собой комплексное решение для управления проектами в организации. Основное назначение системы заключается в автоматизации процессов учета, контроля и анализа проектной деятельности. Система обеспечивает:

- 1) Централизованное хранение информации:
 - полные данные о проектах (наименование, сроки выполнения, этапы реализации);
 - информация о сотрудниках (аналитики, разработчики, начальник) с указанием их ролей и компетенций;
 - информация об отчетах (дата, электронный документ, кроки, статус);
 - история изменений и версий проектов.
- 2) Разграничение прав доступа:
 - трехуровневая система ролей (Аналитик, Программист, Начальник);
 - дифференцированный доступ к функционалу в зависимости от роли;
 - механизм аудита действий пользователей.
- 3) Формирование отчетной документации:
 - автоматическая генерация отчетов в различных форматах;
 - настраиваемые шаблоны отчетных форм;
 - возможность экспорта данных для внешней обработки.
- 4) Интеграционные возможности:
 - подключение к серверным СУБД (MySQL);
 - API для интеграции с другими корпоративными системами.

1.2.2 Эксплуатационное назначение

Программный продукт предназначен для ежедневного использования в рабочих процессах организации. Основные аспекты эксплуатации:

- 1) Режим работы:
 - круглосуточная доступность системы;
 - плановое техническое обслуживание в нерабочие часы;
 - возможность горячего резервирования.
- 2) Пользовательские группы:
 - руководящий состав (Начальники отделов):
 - полный контроль над проектами;
 - управление персоналом;
 - анализ эффективности работы отделов;
 - формирование стратегических отчетов.
 - Технические специалисты (Программисты):
 - ведение проектной документации;
 - отслеживание сроков выполнения задач;
 - взаимодействие с аналитиками;
 - фиксация рабочего времени.
 - Аналитические работники (Аналитики):
 - сбор и систематизация требований;
 - подготовка аналитических материалов;
 - контроль соответствия результатов поставленным задачам;
 - формирование отчетов о ходе работ.
- 3) Результаты работы системы:
 - оперативные отчеты о состоянии проектов;
 - аналитические выкладки по эффективности работы отделов;
 - автоматизированные расчеты премиальных выплат;
 - архив выполненных проектов с возможностью ретроспективного анализа.

1.2.3 Требования к функциональным характеристикам

- 1) Управление пользователями:
 - регистрация новых пользователей с назначением ролей;
 - аутентификация через API;
 - восстановление доступа при утере учетных данных;
 - управление профилями пользователей.
- 2) Управление проектами:
 - создание карточек проектов с указанием:
 - наименования и описания;
 - сроков выполнения (плановых и фактических);
 - ответственных исполнителей;
 - бюджетных показателей.
 - календарное планирование задач;
 - контроль этапов выполнения.
 - система уведомлений о критических изменениях.
- 3) Работа с персоналом:
 - каталог сотрудников с фильтрацией по:
 - отделам;
 - должностям;
 - навыкам;
 - текущей загрузке.
 - назначение ответственных за проекты;
 - учет рабочего времени;
 - анализ загрузки персонала.
- 4) Отчетность и аналитика:
 - стандартные отчетные формы:
 - по проектам;
 - по сотрудникам;
 - финансовые;
 - статистические.
 - пользовательские отчеты с настраиваемыми параметрами;

- визуализация данных (графики, диаграммы);
 - экспорт в различные форматы (PDF, XLSX, DOCX).
- 5) Интеграционные сервисы:
- flask_api для внешних систем;
 - механизм импорта/экспорта данных;
 - сервис очередей для асинхронных задач.

1.2.4 Входные и выходные данные

- 1) Источники входных данных:
 - ручной ввод через пользовательский интерфейс;
 - пакетная загрузка из файлов (JSON, SQL);
 - импорт из внешних систем через API.
- 2) Типы выходных данных:
 - экранные формы с возможностью фильтрации;
 - печатные формы документов;
 - файлы для внешней обработки;
 - уведомления и оповещения;
 - дашборды с ключевыми показателями.
- 3) Форматы данных:
 - внутренний JSON формат для обмена между компонентами;
 - стандартные офисные форматы для отчетности;
 - специализированные форматы для интеграции.

1.2.5 Требования к надежности

- 1) Обеспечение бесперебойной работы:
 - механизм автоматического восстановления после сбоев;
 - репликация критически важных данных;
 - мониторинг работоспособности компонентов.
- 2) Защита данных:
 - шифрование конфиденциальной информации;
 - разграничение прав доступа на уровне данных;
 - журналирование всех значимых событий.
- 3) Резервное копирование:
 - автоматизированный процесс создания бэкапов;
 - хранение нескольких поколений резервных копий;
 - процедуры аварийного восстановления.
- 4) Производительность:
 - оптимизация запросов к базе данных;
 - кэширование часто используемых данных;
 - балансировка нагрузки в пиковые периоды.

1.2.6 Требования к техническим средствам

1) Серверная часть:

- аппаратные требования:
 - процессор: 4+ ядер, 2.5+ GHz;
 - оперативная память: 16+ GB;
 - дисковое пространство: 500+ GB SSD;
 - сетевая карта: 1+ Gbit.
- программное окружение:
 - ОС: Windows 10+;
 - СУБД: MySQL 8.0+;
 - процессор: 2+ ядер, 2+ GHz;
 - оперативная память: 8+ GB.

2) Клиентская часть:

- стационарные рабочие места:
 - ОС: Windows 10+;
 - процессор: 2+ ядер, 2+ GHz;
 - оперативная память: 8+ GB.
- мобильные устройства:
 - ОС: Android 8.0+/iOS 13+;
 - оперативная память: 2+ GB;
 - свободного места: 100+ MB.

1.2.7 Требования к программной совместимости

- 1) Серверные технологии:
 - язык программирования: Kotlin;
 - система сборки: Gradle.
- 2) Клиентские технологии:
 - мобильные платформы: Kotlin;
 - нативные компоненты: Android SDK.
- 3) Форматы данных:
 - обменные: JSON, XML;
 - отчетные: PDF, XLSX, DOCX.
- 4) Протоколы взаимодействия:
 - HTTP/HTTPS.

1.2.8 Техническая документация

- 1) Архитектурные решения:
 - диаграммы компонентов системы;
 - схемы взаимодействия модулей;
 - описание API.
- 2) Руководство администратора:
 - процедуры установки и настройки;
 - методики обслуживания;
 - рекомендации по масштабированию.
- 3) Спецификации баз данных:
 - диаграммы IDEF0, ER;
 - описание таблиц и отношений.

1.2.9 Пользовательская документация

1) Руководство пользователя:

- пошаговые инструкции по основным сценариям (Руководство пользователя);
- описание интерфейсов;
- примеры работы с системой.

2) Справочная система:

- контекстная помощь;

3) Обучающие материалы:

- тестовые сценарии.

1.2.10 Эксплуатационная документация

- 1) Регламенты работы:
 - графики технического обслуживания;
 - процедуры обновления;
 - политики безопасности.
- 2) Методики тестирования:
 - наборы тестовых случаев;
 - чеклисты проверок;
 - сценарии нагрузочного тестирования.
- 3) Планы развития:
 - дорожная карта;
 - календарь обновлений;
 - система учета пожеланий пользователей.

1.3 Обоснование выбора средств реализации программной системы

1.3.1 Анализ требований к клиентской части

- 1) Для мобильного приложения учета проектов требовалось решение, обеспечивающее:
 - широкий охват пользовательской аудитории;
 - стабильную работу на различных устройствах;
 - возможности для дальнейшего масштабирования;
 - поддержку современных стандартов разработки.
- 2) Android была выбрана по следующим причинам:
 - доминирующая доля рынка мобильных ОС (более 70%);
 - гибкость в настройке и кастомизации интерфейса;
 - открытая экосистема с широкими возможностями интеграции;
 - поддержка современных технологий (Kotlin, Jetpack Components).
- 3) Язык программирования Kotlin обеспечивает:
 - полную совместимость с существующей Java-инфраструктурой;
 - современные языковые конструкции (корутины, null-safety);
 - снижение количества шаблонного кода;
 - официальную поддержку Google для Android-разработки;
 - улучшенную производительность по сравнению с Java.

1.3.2 Серверная архитектура системы

- 1) Серверная часть должна была удовлетворять следующим условиям:
 - простота разработки и поддержки;
 - высокая производительность при обработке запросов;
 - гибкость в масштабировании;
 - надежная работа с базами данных;
 - безопасность передаваемых данных.
- 2) Микрофреймворк Flask был выбран благодаря:
 - минималистичному и понятному API;
 - легкости в настройке и развертывании;
 - возможности создания RESTful API;
 - богатой экосистеме расширений;
 - поддержке асинхронных операций.
- 3) использование Python обеспечивает:
 - быструю разработку благодаря простому синтаксису;
 - доступ к мощным библиотекам для работы с данными;
 - кроссплатформенную совместимость;
 - хорошую документацию и сообщество;
 - интеграцию с различными СУБД.

1.3.3 Система хранения данных

1) Критерии выбора СУБД

При выборе системы управления базами данных учитывались:

- надежность и отказоустойчивость;
- производительность при работе с проектами;
- поддержка транзакций;
- возможности резервного копирования;
- совместимость с выбранными технологиями.

2) Обоснование выбора MySQL

Реляционная СУБД MySQL была выбрана потому, что:

- доказанная надежность в production-средах;
- высокая производительность для OLTP-нагрузок;
- поддержка ACID-транзакций;
- широкие возможности настройки и оптимизации;
- хорошая интеграция с Python.

1.3.4 Взаимодействие компонентов системы

1) Схема обмена данными

Архитектура системы построена по принципу:

- клиентское приложение формирует HTTP-запросы;
- сервер обрабатывает запросы и взаимодействует с БД;
- данные передаются в формате JSON;
- реализована система кэширования на клиенте.

2) безопасность передачи данных

Для защиты информации реализованы:

- шифрование передаваемых данных (HTTPS);
- аутентификация по токенам;
- валидация входных параметров;
- защита от SQL-инъекций;
- разграничение прав доступа.

1.3.5 Дополнительные технологии

1) Инструменты разработки

В процессе разработки используются:

- Android Studio – для клиентской части;
- PyCharm – для серверной разработки;
- MySQL – для хранения базы данных.

2) Вспомогательные библиотеки

Для расширения функциональности применяются:

- Retrofit – для сетевых запросов в Android;
- MySQL-Connector – для работы с БД в Python.

2. Разработка программного продукта

2.1 Разработка структуры данных

2.1.1 Модель данных и принципы проектирования

Проектирование структуры базы данных осуществлялось с учетом требований предметной области и предполагаемых сценариев использования системы. Основной задачей было создание гибкой, масштабируемой и производительной модели данных, которая обеспечивала бы эффективное хранение информации о проектах, сотрудниках и отчетности.

При разработке применялся методологический подход, включающий следующие этапы:

- анализ предметной области и выявление ключевых сущностей;
- определение атрибутов для каждой сущности;
- установление связей между сущностями;
- нормализация структуры данных;
- оптимизация для конкретных сценариев использования.

На рисунке 2.1 представлена ER-диаграмма, показывающая структуру базы данных.

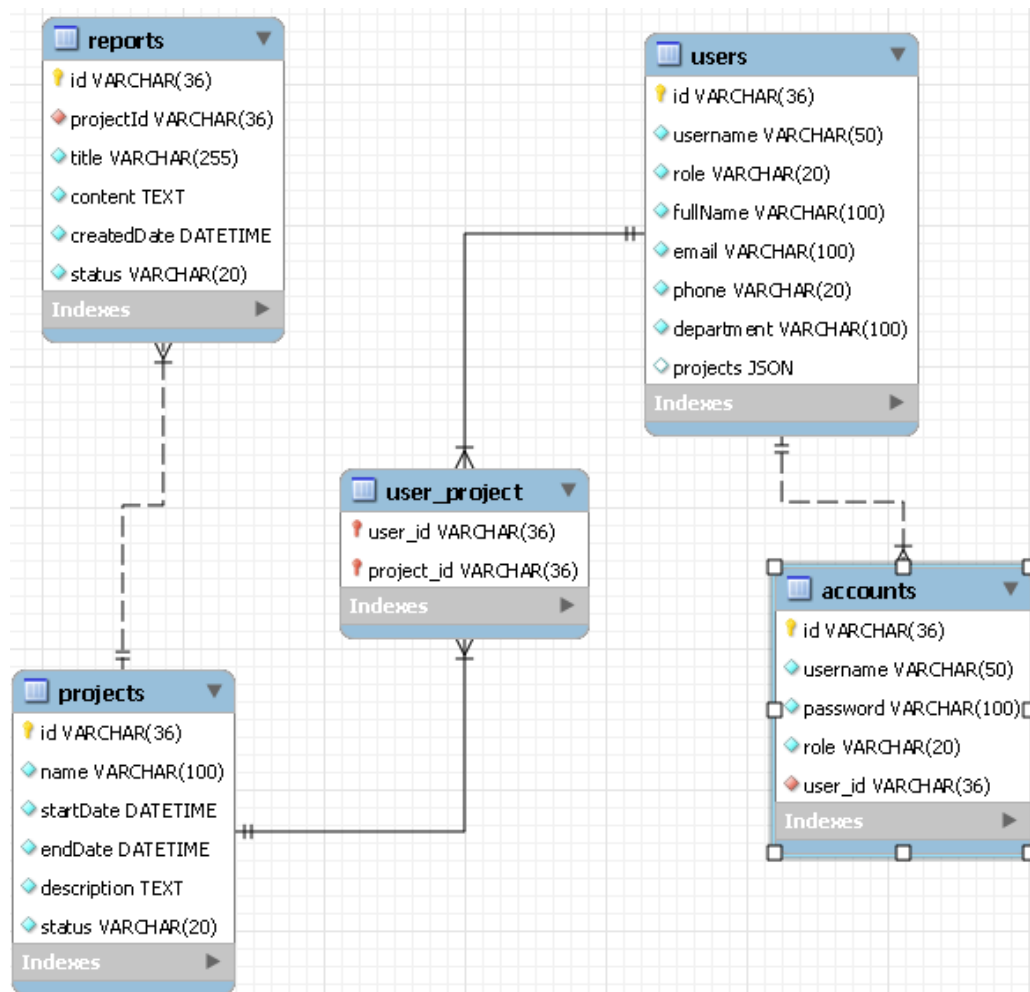


Рисунок 2.1 – ER-диаграмма

2.1.2 Детальное описание сущностей

2.1 Таблица сотрудников организации

Поле	Тип данных	Описание
id	VARCHAR(36)	Идентификатор
username	VARCHAR(50)	Логин пользователя
role	VARCHAR(20)	Роль в системе
fullName	VARCHAR(100)	Полное имя сотрудника
email	VARCHAR(100)	Корпоративная электронная почта
phone	VARCHAR(20)	Контактный телефон
department	VARCHAR(100)	Название отдела/подразделения
projects	JSON	Денормализованный список проектов сотрудника

2.1 Таблица проектов организации

Поле	Тип данных	Описание
id	VARCHAR(36)	Идентификатор
name	VARCHAR(100)	Название проекта
startDate	DATETIME	Фактическая дата начала
endDate	DATETIME	Планируемая дата завершения
description	TEXT	Детальное описание целей и задач
status	VARCHAR(20)	Текущий статус

2.1 Таблица проектных отчетов

Поле	Тип данных	Описание
id	VARCHAR(36)	Идентификатор
projectId	VARCHAR(36)	Ссылка на проект (FK)
title	VARCHAR(255)	Заголовок отчета
content	TEXT	Полное содержание документа
createdDate	DATETIME	Дата и время создания
status	VARCHAR(20)	Статус

2.1 Таблица учетных записей

Поле	Тип данных	Описание
id	VARCHAR(36)	Идентификатор
username	VARCHAR(50)	Логин пользователя
password	VARCHAR(100)	Хеш пароля
role	VARCHAR(20)	Роль в системе
user_id	VARCHAR(36)	Ссылка на сотрудника (FK)

2.1 Таблица назначений на проекты

Поле	Тип данных	Описание
user_id	VARCHAR(36)	Ссылка на сотрудника (FK)
project_id	VARCHAR(36)	Ссылка на проект (FK)

2.2 Схема алгоритма основной программы

На рисунке 2.2 представлена схема алгоритма разделения ролей в программе

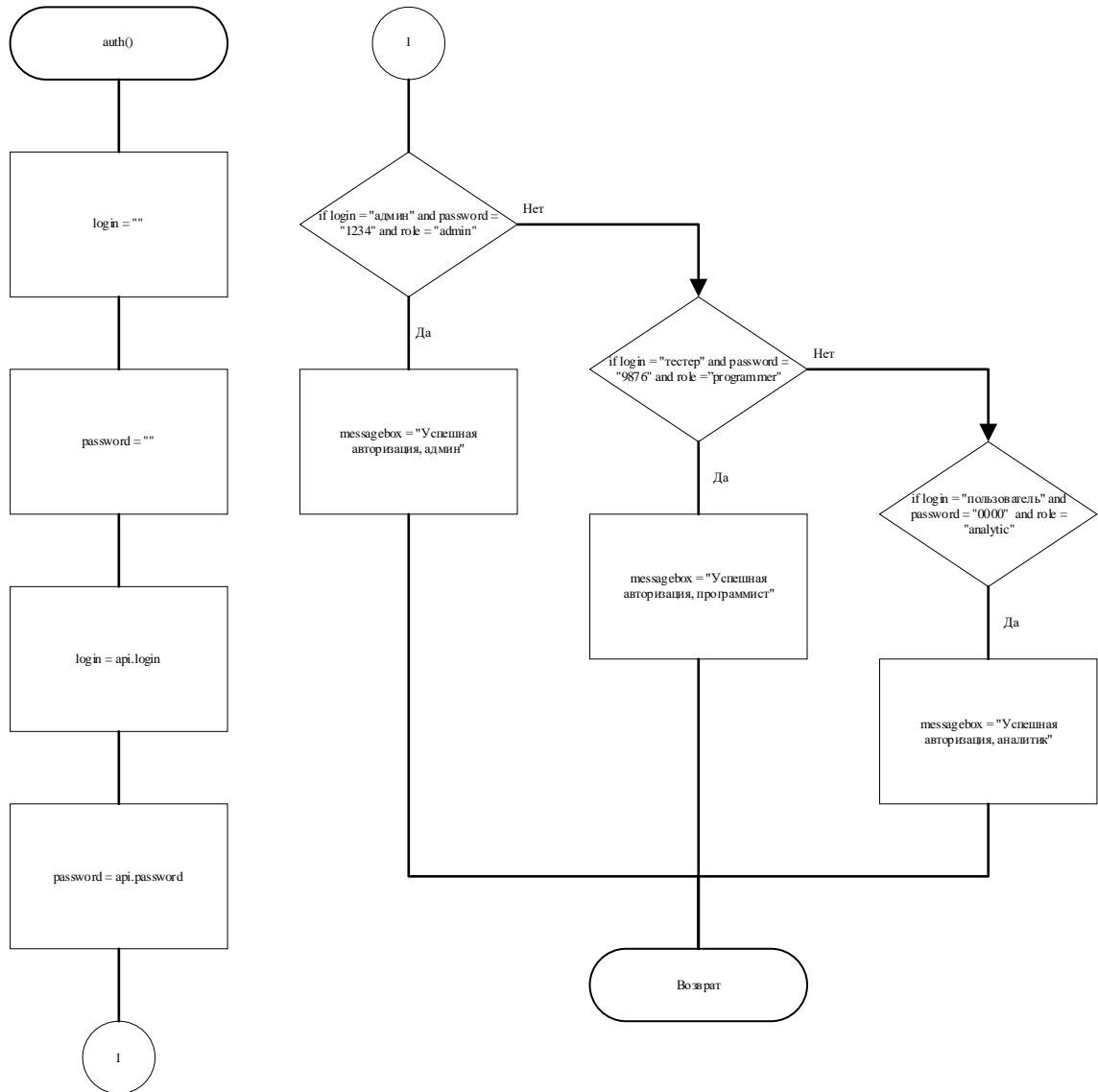


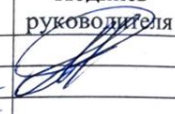



Рисунок 2.2 – Схема алгоритма разделения ролей

На рисунке 2.3 представлена полная структура взаимодействия программ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

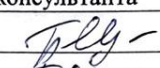
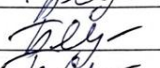
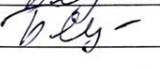
1. Федеральный закон от 27.07.2006 № 149ФЗ «Об информации, информационных технологиях и защите информации» (с изменениями и дополнениями).
 2. Глушаков С.В., Сурядный А.С. Разработка мобильных приложений на Android. — М.: АСТ, 2021.
 3. Джошуа Блох. Java. Эффективное программирование. — СПб.: Питер, 2020.
 4. Советов Б.Я., Цехановский В.В. Базы данных: теория и практика. — М.: Юрайт, 2022.
 5. Иванов А.П. Современные подходы к проектированию кроссплатформенных приложений // Программирование и ИТ. — 2023. — № 4. — С. 45–52.
 6. Петрова Е.К. Использование Kotlin в разработке мобильных приложений // Информационные технологии. — 2022. — № 7. — С. 34–39.
 7. Официальная документация Android Studio [Электронный ресурс]. — Режим доступа: <https://developer.android.com/studio> (дата обращения: 20.03.2025).
 8. Руководство по MySQL Workbench 8.0. — MySQL, 2023.
 9. Public API для данных о футбольных матчах [Электронный ресурс]. — Режим доступа: <https://www.footballldata.org> (дата обращения: 20.03.2025).
- Документация Spring Framework [Электронный ресурс]. — Режим доступа: <https://spring.io/projects/springframework> (дата обращения: 20.03.2025).

Форма 1

Неделя/Дата	Наименование выполненных работ, наблюдения и выводы	Подпись руководителя
1 неделя 24.03-29.03	Исследование предметной области, проведение сравнительного анализа программ-аналогов. Создание функциональной модели предметной области. Моделирование проектируемой системы.	
2 неделя 31.03-5.04	Разработка функциональных требований к программной системе, разработка структурных данных. Разработка кода программного продукта (модуля)	
3 неделя 7.04-12.04	Разработка кода программного продукта (модуля). Испытание существующих способов оптимизации и приемов рефакторинга. Составление схем алгоритмов работы программы. Составление документации	
4 неделя		

6

**График консультаций
на период технологической практики**

№ конс.	Дата недели	Дата конс.	Подпись ст. консультанта	Примечание
1	24.03-29.03	25.03		
2	31.03-5.04	01.04		
3	7.04-12.04	08.04		
4				
5				
6				
7				
8				
9				

Руководитель практики от техникума

Великова О.В.
(Ф.И.О.)Зачет О.Т.Ч.И.О.

8

Отзыв
руководителя практики от предприятия по технологической практике
студента

В ходе технологической практики студент ответственно относился к выполняемой работе, проявлял в повседневном рабочем процессе. Все практику выполнял в срок, стремился к новым знаниям.

Оценка за практику отлично

Руководитель практики
от предприятия

Дата 12.09.18



Ф.И.О. Иванов А.А.

**Аттестационный лист
по производственной практике**

студента (ки) Московского техникума космического приборостроения МГТУ им. Н.Э. Баумана
Дюрягиной Ирины Валерьевны

(Ф.И.О. студента)

Группа **ТИП-82**

Специальность **09.02.07 Информационные системы и программирование Квалификация - Программист**

(код, наименование специальности)

прошел (ла) учебную практику **ПП 01 ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

(наименование практики)

по профессиональному модулю **ПМ.01 Разработка модулей программного обеспечения для компьютерных систем** в объеме 108 часов

(наименование профессионального модуля)

с «24» марта 2025 года по «12» апреля 2025 года

на предприятии (организации) **ПАО «Ил» 125190, г. Москва, пр-т Ленинградский, д. 45г**

(юридический адрес предприятия (организации))

Виды и качество работ в период учебной практики

Виды работ, выполненные студентом во время практики, согласно программы производственной практики	Результат (по 5-ти бальной шкале)
Разработка модулей программного обеспечения для компьютерных систем	5

В ходе учебной практики студентом освоены следующие профессиональные компетенции

Код и название профессиональной компетенции	Результат освоения (освоена/не освоена)
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	Освоена
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.	Освоена
ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.	Освоена
ПК 1.4 Выполнять тестирование программных модулей.	Освоена
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.	Освоена
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.	Освоена

Характеристика профессиональной деятельности обучающегося во время учебной практики: Ирина в течение всей практики ответственно относилась к выполняемой работе, участвовала в повседневной работе. Всю практическую работу выполняла в срок, стремилась к новым знаниям.

Рекомендуемая оценка по практике 5

Руководитель практики от предприятия (организации)

Ил отделе

Должность

Подпись



Ф.И.О. руководителя практики

Итоговая оценка по практике

Руководитель практики от образовательного учреждения

преподаватель

Должность

Подпись

Ф.И.О. руководителя практики