

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)

КУРСОВОЙ ПРОЕКТ

НА ТЕМУ

РАЗРАБОТКА ПОДСИСТЕМЫ

«КОФЕЙНЯ: ЗАКАЗЫ»

Л109. 23КП01. 001 ПЗ

(Обозначение документа)

МДК.02.01 Технология разработки

программного обеспечения

Студент	ИСПП-11	06.12.2024	М.К. Абрамов
	(Группа)	(Подпись)	(И.О. Фамилия)
Преподаватель		06.12.2024	Ю.С. Маломан
		(Подпись)	(И.О. Фамилия)

Архангельск 2024

СОДЕРЖАНИЕ

Перечень сокращений и обозначений.....	3
Введение.....	4
1 Анализ и разработка требований.....	6
1.1 Назначение и область применения.....	?
1.2 Постановка задачи.....	?
1.3 Описание алгоритма функционирования системы.....	?
1.4 Выбор состава программных и технических средств.....	?
2 Проектирование программного обеспечения.....	?
2.1 Проектирование интерфейса пользователя.....	?
2.2 Разработка архитектуры программного обеспечения.....	?
2.3 Проектирование базы данных.....	?
3 Разработка и интеграция модулей программного обеспечения.....	?
3.1 Разработка программных модулей.....	?
3.2 Реализация интерфейса пользователя.....	?
3.3 Разграничение прав доступа пользователей.....	?
3.4 Экспорт и импорт данных.....	?
4 Тестирование и отладка программного обеспечения.....	?
4.1 Структурное тестирование.....	?
4.2 Функциональное тестирование.....	?
5 Инструкция по эксплуатации.....	?
5.1 Установка приложения.....	?
5.2 Инструкция по работе.....	?
Заключение.....	?
Список использованных источников.....	?
Приложение.....	?

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем курсовом проекте применяют следующие сокращения и обозначения:

БД – база данных

ИС – информационная система

ОС – операционная система

ПО – программное обеспечение

СУБД – система управления базами данных

API – интерфейс программирования приложения

ER-модель – модель «сущность-связь»

ERD – диаграмма «сущность-связь»

IDE – интегрированная среда разработки

SQL – язык структурированных запросов

UML – язык структурированных запросов

ВВЕДЕНИЕ

Актуальность разрабатываемого проекта заключается в том, что он решает важную задачу в сфере автоматизации процессов управления заказами в кофейнях.

В условиях современного рынка, где конкуренция среди заведений общественного питания возрастает, эффективное управление заказами становится ключевым фактором успеха. Кофейни сталкиваются с проблемами, связанными с обработкой заказов, взаимодействием с клиентами и оптимизацией рабочего процесса.

Разработка подсистемы для управления заказами в кофейне позволит значительно упростить процесс приема и обработки заказов, улучшить взаимодействие с клиентами и повысить общую эффективность работы заведения.

Целью курсового проектирования является разработка подсистемы обеспечивающей возможность быстрого и удобного оформления заказов, а также управления ими в реальном времени.

Для достижения поставленной цели требуется решить следующие задачи:

- провести сбор и анализ требований целевой аудитории,
- проанализировать информационные источники по предметной области,
- изучить существующие решения в области автоматизации заказов в кофейнях,
- спроектировать архитектуру подсистемы,
- спроектировать диаграмму вариантов использования подсистемы,
- выбрать состав программных и технических средств для реализации проекта,
- спроектировать БД,
- создать БД в выбранной СУБД,

- разработать API для взаимодействия мобильного приложения и телеграм-бота с БД,

- реализовать разграничение прав доступа пользователей,
- обеспечить защиту данных,
- разработать интерфейс мобильного приложения,
- разработать мобильное приложение,
- реализовать функциональность оформления заказа,
- разработать телеграм-бота,
- реализовать функциональность обработки заказов,
- реализовать функциональность управления заказами,
- выполнить структурное тестирование ПО,
- выполнить функциональное тестирование ПО,
- разработать программную документацию,
- разработать эксплуатационную документацию,

В результате выполнения поставленных задач будет создана подсистема «Кофейня: Заказы», которая значительно упростит процесс управления заказами и повысит уровень обслуживания клиентов.

1 Анализ и разработка требований

1.1 Назначение и область применения

Подсистема предназначена для владельцев и сотрудников кофеен, а также для клиентов, желающих упростить процесс оформления и управления заказами. Подсистема позволит пользователям эффективно оформлять, принимать и обрабатывать заказы в реальном времени, а также отслеживать их статус и просматривать информацию о текущих и завершенных заказах.

1.2 Постановка задачи

Необходимо разработать подсистему "Кофейня: Заказы", которая будет включать в себя телеграм-бот для сотрудников кофейни и мобильное приложение для клиентов. Телеграм-бот будет предназначен для обработки заказов, позволяя сотрудникам эффективно управлять процессом выполнения заказов. Мобильное приложение предоставит клиентам возможность просматривать меню, оформлять заказы и отслеживать их статус.

1.3 Описание алгоритма функционирования системы

При запуске мобильного приложения отображается страница меню кофейни, в нижней панели навигации находится иконка пользователя, после нажатия на иконку открывается окно авторизации и регистрации.

При запуске телеграм-бота командой «/start» бот проводит авторизацию сотрудника в системе, после чего открывается функциональность сотрудника в зависимости от его роли в системе.

Клиенты имеют доступ к функциональности, позволяющей им просматривать список товаров, оформлять заказы и просматривать историю

своих заказов. Для доступа к этим возможностям клиентам необходимо пройти авторизацию или регистрацию.

Сотрудники кофейни могут использовать телеграм-бот для просмотра списка новых заказов, а также для отслеживания заказов, находящихся в работе и выполненных. Они могут изменять статус заказов, выбирая соответствующие состояния, такие как "новый", "в работе" или "готов". Кроме того, работники могут выбирать адреса для работы и указывать свое состояние, например, "встать в работу" или "выйти с работы".

Администраторы системы обладают расширенными правами и могут управлять сотрудниками. Они могут добавлять новых сотрудников, удалять существующих, изменять данные и просматривать список всех сотрудников кофейни. Это обеспечивает эффективное управление персоналом и поддержание актуальности информации в системе.

На рисунке 1 изображена диаграмма вариантов использования мобильного приложения.

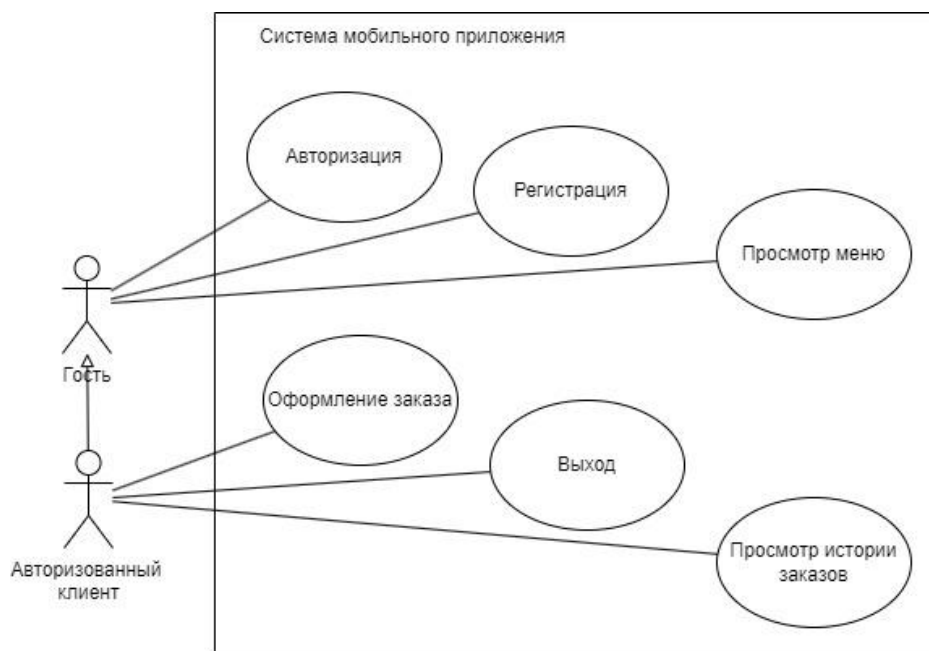


Рисунок 1 – Диаграмма вариантов использования мобильного приложения

На рисунке 2 изображена диаграмма вариантов использования телеграм-бота.

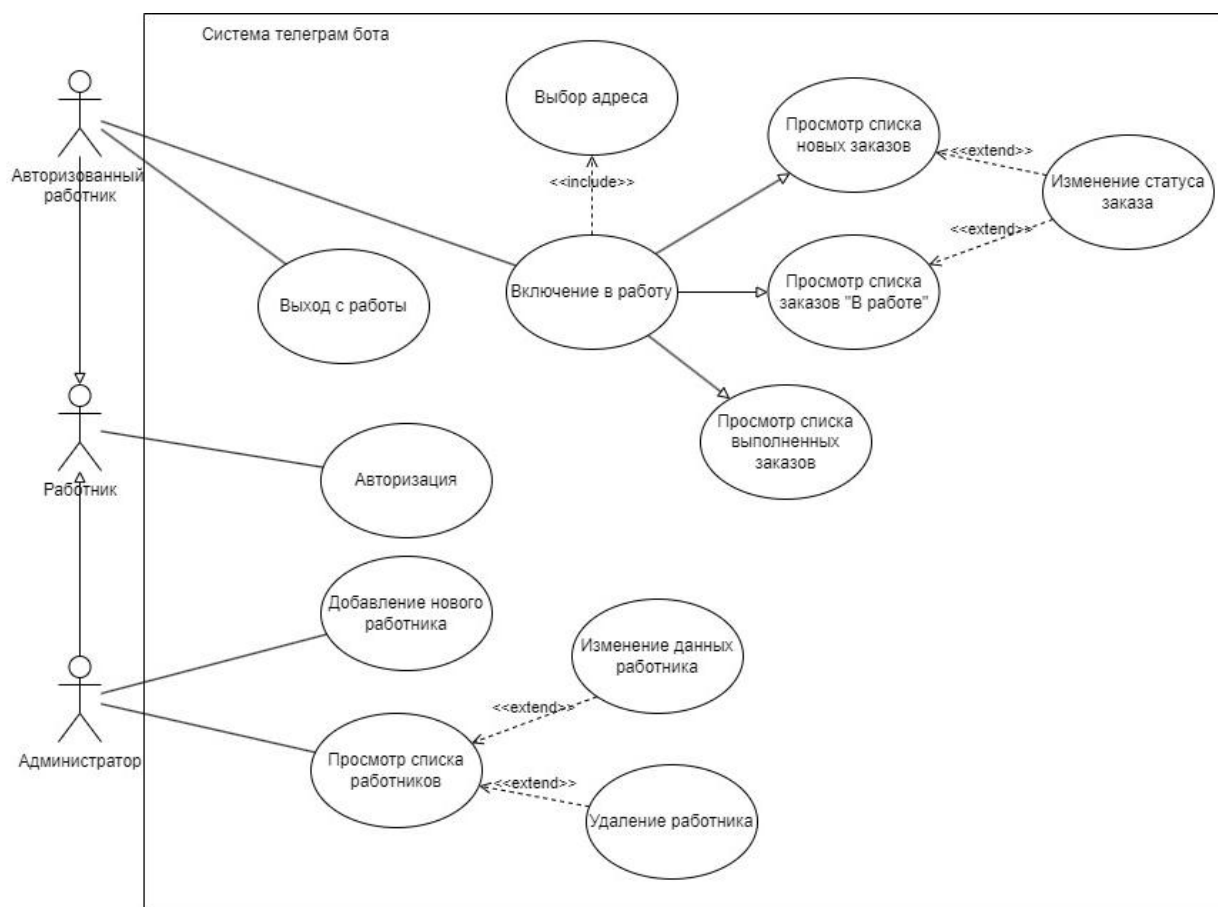


Рисунок 2 – Диаграмма вариантов использования телеграм-бота.

1.4 Выбор состава программных и технических средств

Согласно цели проекта требуется создать подсистему для оформления и обработки заказов.

Работа с мобильным приложением будет осуществляться на мобильных устройствах с установленной операционной системой Android 6.0 или iOS 9.0 и выше с интернет-подключением.

Работа с телеграм-ботом будет осуществляться с помощью приложения «Телеграм» с интернет-подключением.

В качестве СУБД выбрана MySQL 8.0, так как эта СУБД обладает высокой производительностью, простотой в использовании и масштабируемостью, что делает её идеальным выбором для обработки заказов в реальном времени. MySQL также поддерживает множество функций, необходимых для эффективного управления данными, включая транзакции и индексацию.

Мобильное приложение будет написано на языке программирования Dart с использованием фреймворка Flutter, так как он поддерживает кроссплатформенность, обеспечивая высокую производительность и отзывчивость интерфейса. Flutter также предоставляет богатый набор виджетов и инструментов для создания привлекательного пользовательского интерфейса.

Телеграм бот будет разработан на языке программирования Python, так как этот язык поддерживает почти все операционные системы и платформы, имеет много фреймворков и простой синтаксис.

Для разработки мобильного приложения будет использоваться IDE Visual Studio Code, так как она поддерживает множество расширений для работы с Dart и Flutter, а также обеспечивает удобные инструменты для отладки и тестирования.

Для разработки телеграм-бота будет использоваться IDE PyCharm Community Edition 2023.3.4, так как эта среда предлагает удобные инструменты для работы с Python, включая менеджер пакетов и средства отладки.

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

- ОС Ubuntu версии 24 и выше,
- сервер БД: MySQL версии не ниже 8.0,
- процессор с частотой 2 ГГц,

- свободная оперативная память объемом 8 ГБ,
- ПО для конфигурирования, управления и администрирования сервера БД: MySQL Workbench,

- ПО для работы API: dotnet-sdk версии не ниже 8.0,
- ПО для работы телеграм-бота: Python версии не ниже 3.12,

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

- ОС Android версии 6.0 или iOS версии 9.0 и выше,
- процессор с частотой 2 ГГц,
- свободная оперативная память в объеме 2 ГБ,
- свободное место в хранилище 200 МБ,
- постоянное интернет-подключение.

Для функционирования системы на стороне сотрудника достаточны следующие программные и технические средства:

- ОС Android версии 6.0 или iOS версии 9.0 и выше,
- процессор с частотой 2 ГГц,
- свободная оперативная память в объеме 2 ГБ,
- свободное место в хранилище 600 МБ,
- постоянное интернет-подключение,
- приложение «Телеграм».

2 Проектирование ПО

2.1 Проектирование интерфейса пользователя

Изображения каркасов (wireframe) и/или мокапов представлены на рисунке(ах) **x**

Выбранные цвета: белый, черный, #EC5E32

Шрифт: **HereTheSuperInterestingFont**

Иконки представлены на рисунке **x**.



Рисунок **x** - draw.io. Иконки мобильного приложения

2.1 Разработка архитектуры программного обеспечения

2.3 Проектирование базы данных

Требуется разработать БД для системы оформления и обработки заказов кофеен. Система будет использоваться владельцами и сотрудниками кофеен, а также клиентами этих заведений.

Модели БД созданы при помощи MySQL Workbench. На рисунке **?** в виде ERD показана физическая модель предметной области.

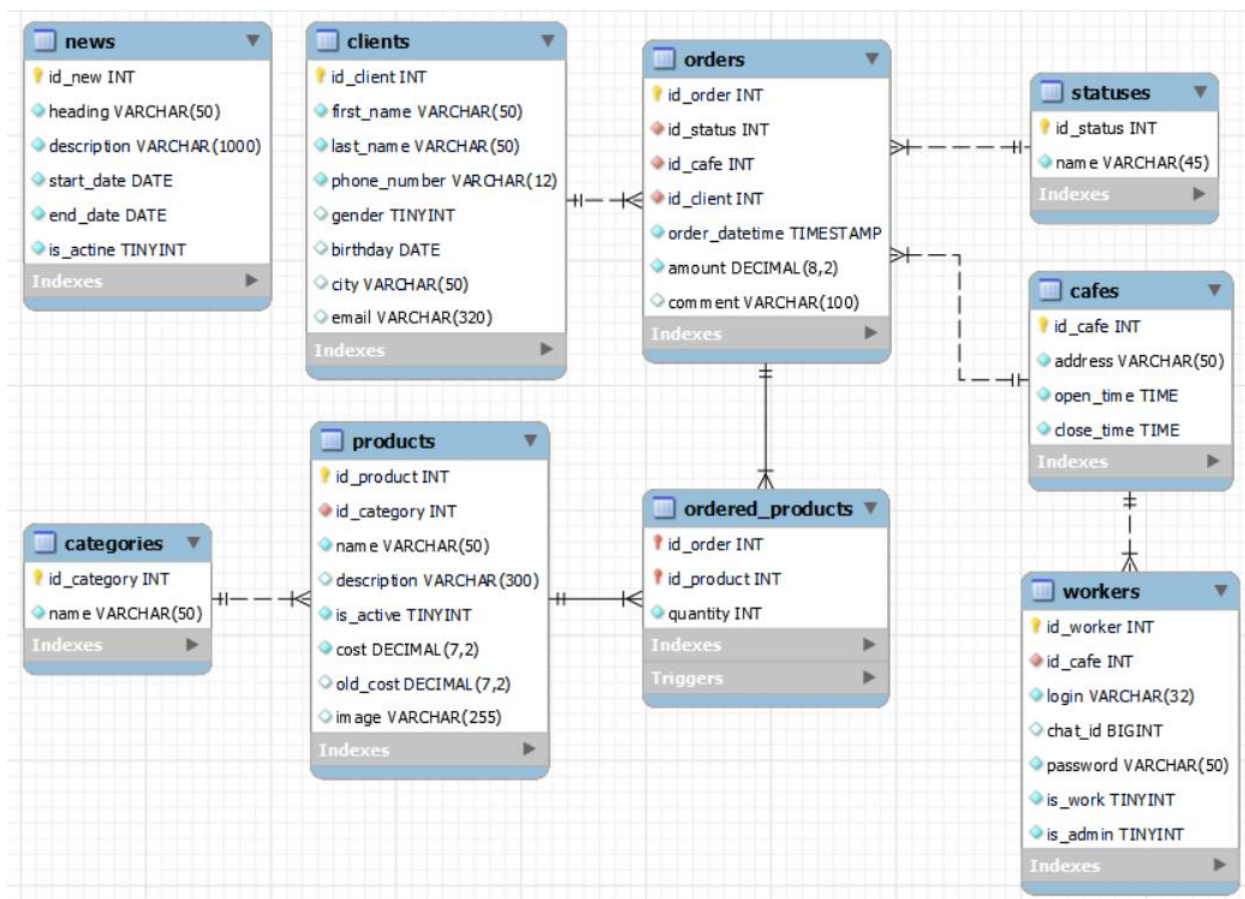


Рисунок 3 – MySQL Workbench. Физическая модель

В таблице 1 в виде словаря данных представлено описание созданных таблиц и ограничений целостности БД.

Таблица 1 – Словарь данных

Ключ	Поле	Тип данных	Обязательное	Примечание
clients				
PK	id_client	INT	+	AI
	first_name	NVARCHAR(50)	+	
	last_name	NVARCHAR(50)	+	
	phone_number	VARCHAR(12)	+	UQ
	gender	BOOL	-	
	birthday	DATE	-	
	city	NVARCHAR(50)	-	

	email	VARCHAR(320)	-	
orders				
PK	id_order	INT	+	AI
FK1	id_status	INT	+	
FK2	id_cafe	INT	+	
FK3	id_client	INT	+	
	order_datetime	TIMESTAMP	+	Значение по умолчанию: CURRENT_ TIMESTTAMP
	amount	DECIMAL(8,2)	+	По умолчанию: 1.00
	comment	NVARCHAR(100)	-	
statuses				
PK	id_status	INT	+	AI
	name	NVARCHAR(50)	+	
cafes				
PK	id_cafe	INT	+	AI
	address	NVARCHAR(50)	+	
	open_time	TIME	+	По умолчанию: 07:00
	close_time	TIME	+	По умолчанию: 20:00
workers				
PK	id_worker	INT	+	AI
FK	id_cafe	INT	+	
	login	VARCHAR(32)	+	
	chat_id	BIGINT	-	
	password	VARCHAR(50)	+	
	is_work	BOOL	+	По умолчанию: False
	is_admin	BOOL	+	По умолчанию: False
ordered_products				
PK	id_order	INT	+	
FK1				
PK	id_product	INT	+	
FK1				
	quantity	INT	+	По умолчанию: 1

products				
PK	id_product	INT	+	AI
FK	id_caregory	INT	+	
	name	NVARCHAR(50)	+	
	description	NVARCHAR(300)	-	
	is_active	BOOL	+	По умолчанию: True
	cost	DECIMAL(7,2)	+	По умолчанию: 0.00
	old_cost	DECIMAL(7,2)	-	
	image	VARCHAR(255)	-	
categories				
PK	id_caregory	INT	+	AI
	name	NVARCHAR(50)	+	

3 Разработка и интеграция модулей программного обеспечения

3.1 Разработка программных модулей

В ходе курсового проектирования были разработаны: мобильное приложение на Dart, телеграм-бот на Python, Web-API приложение на C#.

Для постоянного получения новых заказов в телеграм-боте была разработана функция представленная листингом **x**.

Получение информации из БД осуществляется посредством Web-API приложения, код получения заказа по его ID представлен листингом **x+1**.

Листинг **x** - Код функции непрерывного получения новых заказов

```
async def monitor_new_orders(msg: Message):
    """ Мониторит новые заказы и отправляет сообщения об этих
    заказах пользователю. :param msg: Сообщение от пользователя.
    """
    logging.info("Посещение функции monitor_new_orders")
    worker = await APIMethods.get_worker_state(msg.chat.id)
    while worker:
        worker_address_id = await
APIMethods.get_worker_address(msg.chat.username.replace("_",
"\_"))
        worker_address_str = await
APIMethods.get_address_dict()
        worker_address_str =
worker_address_str.get(worker_address_id)
        new_orders_items = await
APIMethods.monitor_new_orders(worker_address_str) # Метод для
получения новых заказов
        for order_id, order_msg in new_orders_items.items():
            await asyncio.sleep(0.5) # Задержка между отправками
сообщений
            order_keyboard = await orders_inline_keyboard(order_id)
            # Кнопки для обновления статуса заказа
            await msg.answer(order_msg, reply_markup=order_keyboard)
            await asyncio.sleep(25) # Задержка перед новым запросом
```

Листинг **x+1** - Код получения заказа по его ID

```
// GET: api/order/5
[HttpGet("info/id={id}")]
public async Task<ActionResult<orders_full_info_view>>
GetOrderInfo(int id)
{
    var order = await
_context.orders_full_info_views.FindAsync(id);
    if (order == null){return NotFound();}
    return order;
}
```

3.2 Реализация интерфейса пользователя

Интерфейс мобильного приложения разработан с использованием постраничной навигации, в приложении разработаны различные элементы управления, стили и виджеты для упрощения работы. Навигация в приложении реализована с помощью виджета Navigator, который управляет стеком объектов Route, представляющих страницы в приложении.

Для отображения информации об экспертах разработан виджет SearchExpertItem, который представлен листингом 3.

Листинг 3 – Код виджета для SearchExpertItem

3.3 Разграничение прав доступа пользователей

Разграничение прав доступа реализовано разделением сотрудников и клиентов по разным таблицам. Права администратора у сотрудника появляются при True значении в поле is_admin.

Пример кода раграничения прав пользователей телеграм бота представлен листингом x.

Листинг x - Код раграничения прав пользователей телеграм бота


```

async def keyboard_menu(msg: Message) -> ReplyKeyboardMarkup:
    """
        Создает клавиатуру для главного меню.
        :return: Объект ReplyKeyboardMarkup с кнопками меню.
    """
    # Определяем кнопки для клавиатуры
    if await
    APIMethods.get_admin_state(msg.chat.username.replace("_",
    "\_")):
        keyboard_list = [[KeyboardButton(text=variables.new_worker)],
        [KeyboardButton(text=variables.read_workers)], [KeyboardButton
        (text=variables.del_worker)]]
    else:
        keyboard_list = [[KeyboardButton(text=variables.address),
        KeyboardButton(text=variables.work)], [KeyboardButton(text=var
        iables.home), KeyboardButton(text=variables.complete)]]
    # Создаем и возвращаем объект клавиатуры
    keyboard = ReplyKeyboardMarkup(keyboard=keyboard_list,
        resize_keyboard=True, input_field_placeholder="Выберите
        действие:")
    return keyboard

```

3.4 Экспорт и импорт данных

Считается ли регистрация за импорт? Или изменение данных пользователей системы?

4 Тестирование и отладка ПО

4.1 Структурное тестирование

Тестирование белым ящиком телеграм бота

4.2 Функциональное тестирование

Тестирование черным ящиком бота/мобилки

5 Инструкция по эксплуатации ПО

5.1 Установка программного обеспечения

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

- ОС Ubuntu версии 24 и выше,
- процессор с частотой 2 ГГц,
- свободная оперативная память объемом 8 ГБ,
- свободное место на диске – 10 ГБ;
- дополнительные компоненты: dotnet-sdk версии не ниже 8.0, Python версии 3.12 или выше,
- наличие интернет-соединения,
- MySQL Server 8.0,
- Docker,
- docker-compose.

Процесс создания БД:

- авторизация на сервере, в СУБД,
- выполнить импорт / запуск скрипта базы данных CoffeShopDB.sql.

Для функционирования мобильного приложения на стороне клиента достаточны следующие программные и технические средства:

- ОС Android версии 6.0 или iOS версии 9.0 и выше,
- процессор с частотой 2 ГГц,
- свободная оперативная память в объеме 2 ГБ,
- свободное место в хранилище 200 МБ,
- постоянное интернет-подключение.

Для функционирования системы на стороне сотрудника достаточны следующие программные и технические средства:

- ОС Android версии 6.0 или iOS версии 9.0 и выше,

- процессор с частотой 2 ГГц,
- свободная оперативная память в объеме 2 ГБ,
- свободное место в хранилище 600 МБ,
- постоянное интернет-подключение,
- приложение «Телеграм».

Для установки мобильного приложения требуется выбрать собранный **ОченьКрутоеНазвание**.apk файл с приложением в проводнике и установить его.

В мобильном приложении используются следующие данные для авторизации:

- Номер телефона – 88005553535:
- Электронная почта – Ivan@mail.ru.

В телеграм-боте:

- Логин – задан по умолчанию как логин телеграм;
- Пароль – passwd.

5.2 Инструкция по работе

Для запуска мобильного приложения “Name” требуется нажать на иконку этого приложения

Процесс авторизации работников в телеграм боте:

- Открыть чат с ботом
- Отправить боту команду “/start”
- Бот установит user_name пользователя как логин, попросит ввести пароль
- Отправить пароль, полученный ранее от администратора
- Получить сообщение об успешной авторизации

- В зависимости от заранее предустановленной настройки ролей происходит вход в систему от имени работника или от имени администратора

Процесс регистрации клиентов в мобильном приложении:

- открытие приложения,
- ввод адреса электронной почты,
- Получение кода авторизации на почту
- Ввод кода авторизации, отправка кода в приложении
- Ввод персональных данных пользователя
- Нажать кнопку подтверждение данных

Возможности пользователей:

ЗАКЛЮЧЕНИЕ

В ходе курсового проектирования достигнута поставленная цель: разработана подсистема «Кофейня: Заказы».

разработка подсистемы обеспечивающей возможность быстрого и удобного оформления заказов, а также управления ими в реальном времени.

Разработанная подсистема поможет обеспечить возможность быстрого и удобного оформления заказов, а также управления ими в реальном времени.

Кроме того, решены все поставленные задачи:

- выполнен сбор и анализ требований целевой аудитории,
- проанализированы информационные источники по предметной области,
- изучены существующие решения в области автоматизации заказов в кофейнях,
- спроектирована архитектура подсистемы,
- спроектирована диаграмма вариантов использования подсистемы,
- выбран состав программных и технических средств для реализации проекта,
- спроектирована физическая схема БД,
- разработано API для взаимодействия мобильного приложения и телеграм-бота с БД,
- реализовано разграничение прав доступа пользователей,
- реализована защита данных,
- разработано мобильное приложение,
- разработан телеграм-бот,
- выполнено структурное тестирование ПО,
- выполнено функциональное тестирование ПО,
- разработана программная документация,
- разработана эксплуатационная документация,

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голицына, О. Л. Базы данных : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 4-е изд., перераб. и доп. – Москва : ФОРУМ : ИНФРА-М, 2020. – 400 с. – URL: <https://znanium.com/catalog/document?id=362825>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
2. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. – 2-е изд., перераб. и доп. Москва : ФОРУМ : ИНФРА-М, 2021. – 416 с. – URL: <https://znanium.com/catalog/document?id=364900>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
3. Дадян, Э. Г. Данные: хранение и обработка : учебник / Э. Г. Дадян. – Москва : ИНФРА-М, 2020. – 205 с. – URL: <https://znanium.com/catalog/document?id=346013>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
4. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем : Учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2018. – 368 с. – URL: <http://znanium.com/bookread2.php?book=926871>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
5. Павловская, Т. А. С#. Программирование на языке высокого уровня: Учебник для вузов / Т. А. Павловская. – Санкт-Петербург : Питер, 2021. – 432 с. – URL: <https://ibooks.ru/bookshelf/377952/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
6. Туманов, В. Е. Основы проектирования реляционных баз данных / В. Е. Туманов. – Москва : Национальный Открытый Университет «ИНТУИТ», 2020. – 502 с. – URL: <https://www.iprbookshop.ru/97570.html>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

7. Фленов, М. Е. Библия С# / М. Е. Фленов. – 4-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2019. – 512 с. – URL: <https://ibooks.ru/bookshelf/366634/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.