|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сергеева Е.Г.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |

**Руководство программиста**

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | СОГЛАСОВАНО |
| Колледж ВятГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Беляев И.Д.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |  | Преподаватель УП.03  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М.Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

2024

Содержание

[1 Общие сведения о программе 3](#_Toc182341892)

[1.1 Назначение программы 3](#_Toc182341893)

[1.2 Функции программы 3](#_Toc182341894)

[1.3 Минимальный состав технических средств 3](#_Toc182341895)

[1.4 Минимальный состав программных средств 4](#_Toc182341896)

[1.5 Требования к персоналу (системному программисту) 4](#_Toc182341897)

[2 Структура программы 5](#_Toc182341898)

[2.1 Сведения о структуре программы 5](#_Toc182341899)

[2.2 Сведения о составных частях программы 5](#_Toc182341900)

[2.3 Сведения о связях между составными частями программы 5](#_Toc182341901)

[2.4 Сведения о связях с другими программами 5](#_Toc182341902)

[3 Настройка программы 6](#_Toc182341903)

[3.1 Настройка на состав технических средств 6](#_Toc182341904)

[3.2 Настройка на состав программных средств 6](#_Toc182341905)

[3.2.1 Установка информационной системы 6](#_Toc182341906)

[3.2.2 Настройка базы данных 6](#_Toc182341907)

[4 Установка программы 7](#_Toc182341908)

[4.1 Установка серверной части 7](#_Toc182341909)

[4.2 Установка клиентской части 7](#_Toc182341910)

[5 Проверка программы 8](#_Toc182341911)

[5.1 Описание способов проверки 8](#_Toc182341912)

[5.2 Методы прогона 8](#_Toc182341913)

[6 Сообщения системному программисту 9](#_Toc182341914)

[6.1 Сообщения об ошибках 9](#_Toc182341915)

[6.1.1 Ошибка подключения к серверу или базе данных 9](#_Toc182341916)

[6.1.2 Ошибка аутентификации и авторизации 9](#_Toc182341917)

[6.1.3 Ошибка в работе с данными 9](#_Toc182341918)

[6.1.4 Сетевая ошибка 10](#_Toc182341919)

Аннотация

В данном программном документе приведено руководство программиста по установке и настройке программного обеспечения, информационной системы учета животных в приютах.

В данном программном документе выделяется пять разделов. В первом разделе «Общие сведения о программе» указаны функции приложения, сведения о программных и технических средствах, требования к персоналу.

Во втором разделе «Структура программы» приведены сведения о структуре приложения, её составных частях, кроме того, в данном разделе описываются связи между составными частями программы.

В третьем разделе «Настройка программы» описаны действия по настройке приложения с использованием технических и программных средств.

В четвертом разделе «Установка программы» описаны действия по установке приложения на клиентские и пользовательские технические устройства.

В пятом разделе «Проверка программы» приведено описание способов проверки, для определения работоспособности информационной системы.

В шестом разделе «Сообщения системному программисту» указаны тексты сообщений, полученных в ходе компиляции программы, содержащие в себе описание действий, которые необходимо предпринять следуя выведенным сообщениям.

Данная программная документация оформлена по требованиям ГОСТ 19.101-77, ГОСТ 19.103-77, ГОСТ 19.104-78, ГОСТ 19.503-79.

# Общие сведения о программе

## Назначение программы

Информационная система для кинотеатра, предназначена для автоматизации управления кинотеатром, упростить и автоматизировать рутинные операции, такие как продажа билетов, бронирование мест и управление сеансами. Система позволит автоматизировать работу кинотеатра, автоматизировать процессы учета продажи билетов.

## Функции программы

* Вывод сводных отчетов в формате «.pdf» в информационную систему, доступны следующие отчеты: «Отчет о фильмах», «Отчет о клиентах, «Отчет о продажах билетов»;
* Добавление пользователей в систему, назначение ролей пользователям, для выделения и предоставления конкретного функционала необходимого для работы, конкретного пользователя;
* Формирование отчетности для изучения актуальности работоспособности, деятельности предприятия (кинотеатра);
* Добавление, нового фильма в систему;
* Добавление нового фильма в каталоге для подробного ознакомления пользователя не являющимся работником со списком фильмов, список в каталоге создает пользователь, являющийся администратором;
* Вывод информации о фильме в каталог, приложение отображает из базы данных фильмов в каталоге, предоставляет возможность пользователям, не являющимися работниками просмотреть доступные фильмы при помощи каталога;

## Минимальный состав технических средств

Минимально необходимый состав технических средств:

* Процессор с тактовой частотой не менее 1,0 ГГц.
* Оперативная память, объемом не менее 1 Гб.
* Внутренняя память, объемом не менее 64 Гб с возможностью расширения через microSD.
* Емкость аккумулятора не менее 4000 мАч.
* Поддержка сетей 4G и Wi-Fi 802.11ac.
* Bluetooth версии не ниже 5.0.

## Минимальный состав программных средств

Для функционирования информационной системы для кинотеатра, необходим следующий состав программных средств:

* Операционная система Android версии не ниже 7.0.
* Android Studio – основная среда разработки, включающая эмулятор, инструменты для отладки и профилирования.
* Android SDK – комплект разработчика для Android, содержащий необходимые библиотеки и утилиты.
* Flutter – фреймворк необходимый для разработки Android приложений.
* API для связи с сервером – библиотека, такая как node.js, для взаимодействия с REST API.
* PostgreSQL (на стороне сервера) – основная база данных для хранения данных. Прямое подключение к PostgreSQL с Android не рекомендуется, вместо этого используется REST API для обмена данными с сервером.

## Требования к персоналу (системному программисту)

Системный программист должен уверенно владеть фреймворком Flutter, знать и уверенно владеть таким языком программирования как Dart, JavaScript для работы с серверной частью, реализации REST API. Кроме того, системный программист должен иметь знания в области реляционных баз данных, а именно PostgreSQL, включая разработку схем данных, написание SQL запросов.

Системный программист должен поддерживать работоспособность технических средств, должен выполнять задачи связанные с инсталляцией и поддержкой работоспособности системных программных средств, поддерживать работоспособность информационной системы для кинотеатра.

# Структура программы

## Сведения о структуре программы

Структура программы организована по модульному принципу. Приложение состоит из серверной части, клиентской части, базы данных, а также API для взаимодействия между ними.

## Сведения о составных частях программы

Клиентская часть Android разработана с использованием фреймворка Flutter для взаимодействия пользователей через мобильное приложение.

Серверная часть разработана на JavaScript с использованием node.js. Необходима для управления пользователями, учета животных, управления заявками, ведения задач кормления, уборки, лечения.

База данных используется PostgreSQL для хранения информации о пользователях, животных, заявках, задачах.

API для взаимодействия между компонентами, REST API позволяет клиентским приложениям обмениваться данными с сервером при помощи различных маршрутов.

## Сведения о связях между составными частями программы

Фронтенд часть взаимодействует с бэкенд частью через REST API, отправляя HTTP-запросы для получения или отправки данных.

Бэкенд часть обрабатывает запросы, выполняет необходимые операции с базой данных и возвращает ответ фронтенд части.

База данных хранит и предоставляет данные бэкенд части при каждом запросе.

## Сведения о связях с другими программами

Связь между информационной системой учета животных в приютах и другим программным обеспечением отсутствует.

# Настройка программы

## Настройка на состав технических средств

Для корректной работы информационной системы необходимо убедиться, что устройство соответствует необходимому минимальному составу технических средств, настроить сетевое подключение, проверить стабильность соединения.

## Настройка на состав программных средств

### Установка информационной системы

Для начала установки необходимо скачать APK-файл приложения из внутреннего репозитория приюта, далее необходимо установить приложение на устройство, убедившись, что версия Android соответствует минимальному требованию, проверить что приложение имеет права на доступ к сети для связи с сервером.

### Настройка базы данных

Для начала необходимо установить PostgreSQL на сервере, предназначенном для хранения данных, далее необходимо создать базу данных и таблицы, необходимые для хранения необходимой информации, согласно структуре системы. Также необходимо проверить стабильность соединения между сервером и базой данных, заполнить базу данными, убедиться в корректности сохранения и доступности данных, настроить регулярное резервное копирование базы данных.

# Установка программы

## Установка серверной части

Для начала необходимо скачать и установить node.js последней версии с официального сайта[[1]](#footnote-1), далее необходимо установить менеджер пакетов «npm» и необходимые зависимости для работы с серверной частью, такие как «Express.js», библиотеку для работы с базой данных «pg».

Далее необходимо произвести настройку переменных окружения, создать файл «.env» для хранения конфиденциальной информации базы данных, внести в файл необходимые настройки.

Также необходимо установить PostgreSQL на сервере, создать базу данных.

Далее необходимо запустить серверную часть при помощи команды «node server.js», проверить подключение.

## Установка клиентской части

Для начала необходимо скачать APK-файл приложения из внутреннего репозитория приюта, далее необходимо установить приложение на устройство, убедившись, что версия Android соответствует минимальному требованию, проверить подключение к сети и серверу.

# Проверка программы

## Описание способов проверки

Для проверки работоспособности информационной системы можно использовать следующие методы:

Модульное тестирование (Unit Testing), используется для проверки необходимых компонентов программы, функций. Для тестирования можно использовать модульные тесты C# с использованием библиотеки xUnit.

* Функциональное тестирование - Бронирование билетов: Проверка возможности выбора фильма, времени сеанса, места в зале и оплаты. Управление расписанием: Тестирование добавления, изменения и удаления сеансов. Управление пользователями: Проверка регистрации, авторизации и восстановления пароля. Отчеты и аналитика: Проверка генерации отчетов по продажам, посещаемости и другим метрикам. Интеграция с платежными системами: Тестирование корректности оплаты и возврата средств.

Интеграционное тестирование, тип тестирования который проверяет взаимодействие между сервером и базой данных, клиентом и сервером. Необходимо убедиться, что компоненты системы работают в связке корректно без возникновения ошибок.

Тестирование производительности, оценка работоспособности системы при высокой нагрузке, проверка сервера при поступлении большого количества запросов, большого объема данных.

Тестирование пользовательского интерфейса, проверка интерфейса приложения на различных Android устройствах с различным разрешением экрана. Функциональное тестирование UI интерфейса, проверка корректности отображения.

## Методы прогона

Прогон включает в себя тестирование информационной системы на реальных данных и проверку в различных условиях.

Для прогона информационной системы можно использовать следующие методы:

Автоматические тесты API, прогон API запросов с помощью инструментов, таких как Postman, чтобы убедиться что сервер корректно обрабатывает запросы, взаимодействует с базой данных и возвращает корректные ответы.

Ручное тестирование функциональности, проверка ключевых функций программы вручную. Добавление новых пользователей, животных, создание заявок, редактирование данных, удаление записей и другие операции.

# Сообщения системному программисту

При эксплуатации информационной системы учета животных в приютах могут возникать различные ошибки. Например ошибки конфигурации, ошибки взаимодействия с пользователями, аппаратные или сетевые сбои. Такие ошибки оказывают влияние на работоспособность системы и требуют вмешательства системного программиста для их устранения.

## Сообщения об ошибках

### Ошибка подключения к серверу или базе данных

Возникает ошибка при подключении «Connection timed out.». Программа не может подключиться к серверу базы данных из-за проблем с сетью или сбоя в работе сервера.

Причинами ошибки могут оказаться, неправильная конфигурация подключения, отключение базы данных или сервера, проблемы с сетевым соединением.

Для устранения ошибки необходимо проверить конфигурацию, убедиться что параметры подключения к базе данных правильные, проверить работоспособность сервера базы данных для этого можно использовать команды «ping», «telnet». Если сервер базы данных не отвечает, можно попробовать его перезапустить. Если проблема связана с сетью, можно проверить состояние соединения.

### Ошибка аутентификации и авторизации

Возникает ошибка при авторизации «Invalid username or password.», пользователь не может войти в систему, даже при наличии правильных учетных данных.

Причинами ошибки могут оказаться неверные или устаревшие учетные данные пользователей, проблемы с системой токенов или сессий.

Для устранения ошибки необходимо убедиться, что используемые логин и пароль корректны, проверить не истек ли срок действия сессии пользователя, если сессия истекла необходимо её перезагрузить или выполнить повторную авторизацию.

### Ошибка в работе с данными

Возникает ошибка при обработке данных «DataProcessingError: Unable to process data. Invalid input format detected for 'phone number'.», программа не правильно обрабатывает или отображает данные.

Причиной ошибки могут оказаться, неверные SQL запросы, приводящие к получению некорректных или неполных данных, проблемы с кодировкой данных или форматом ввода, ошибки валидации данных.

Для устранения ошибки необходимо протестировать SQL запросы вручную с использованием SQL клиента PostgreSQL, убедиться что запросы возвращают правильные результаты. Убедиться, что входные данные соответствуют ожидаемому формату. Можно добавить валидацию на уровне клиента. Чтобы избежать проблем с инъекциями необходимо использовать подготовленные запросы.

### Сетевая ошибка

Возникает ошибка при возникновении проблем с сетью «Unable to connect to the server.». Данная ошибка влияет на обмен данными между клиентом и сервером.

Причинами могут служить, потеря соединения с сервером или медленное соединение, проблемы с настройкой DNS или прокси, ошибка маршрутизации запросов.

Для устранения ошибки необходимо использовать команды «ping», «telnet», другие инструменты для диагностики сетевых проблем. Необходимо убедиться, что порты для нужных сервисов открыты, для повышения надежности необходимо использовать резервные серверы и механизмы восстановления соединения.

перечень терминов и сокращений

API - Application Programming Interface (Интерфейс программирования приложений)

SDK - Software Development Kit (Набор для разработки программного обеспечения)

SQL - Structured Query Language (Язык структурированных запросов)

DNS - Domain Name System (Система доменных имен)

1. Официальный сайт node.js - https://nodejs.org/en [↑](#footnote-ref-1)