|  |  |  |
| --- | --- | --- |
|  |  | МНОГОПОЛЬЗОВАТЕЛЬСКАЯ СИСТЕМА ЗАПИСИ ДОЛГОВ |
|  |  | Долговая яма |
|  |  | Технический проект |
|  |  | А.В.00001-01 33 01-1 |
|  |  | (вид носителя данных) |
|  |  | Листов 13 |
| **Подп. И дата** |  |  |
| **Инв. №дубл.** |  |  |
| **Взам. Инв. №** |  |  |
| **Подп. И дата** |  |  |
| **Инв. №подл.** |  |  |

2 0 2 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | СОГЛАСОВАНО | | УТВЕРЖДАЮ | |
|  |  | Заказчик | | Начальник управления | |
|  |  | Личная  подпись | Расшифровка  подписи | Личная  подпись | Расшифровка  Подписи |
|  |  | 15.04.23 | | 15.04.23 | |
|  |  | МНОГОПОЛЬЗОВАТЕЛЬСКАЯ СИСТЕМА ЗАПИСИ ДОЛГОВ | | | |
|  |  | Долговая яма | | | |
| ***Подп. И дата*** |  | Технический проект | | | |
| ЛИСТ УТВЕРЖДЕНИЯ | | | |
| А.В.00001-01 33 01-1-ЛУ | | | |
| ***Инв. № дубл.*** |  | (вид носителя данных) | | | |
| Листов 13 | | | |
| ***Взам. Инв.№*** |  | СОГЛАСОВАНО | | Представители  Предприятия-разработчика | |
| Руководитель | | Руководитель разработки | |
| ***Подп. И дата*** |  | Личная  подпись | Расшифровка  подписи | Личная  подпись | Расшифровка  подписи |
| 15.04.23 | | 15.04.23 | |
|  | |  | |
| ***Инв.№подл.*** |  |  |  |  |  |
|  | |  | |
|  |  |  |  |  | |
|  |  |  |  |  |  |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  |  |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  |  |

2 0 2 3

**СОДЕРЖАНИЕ**

[1 ВВЕДЕНИЕ 4](#_Toc132920579)

[1.1 Полное наименование программы 4](#_Toc132920580)

[1.2 Краткое наименование программы 4](#_Toc132920581)

[1.3 Назначение и область применения 4](#_Toc132920582)

[1.4 Требования к функционалу 4](#_Toc132920583)

[2 СТРУКТУРА ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ 5](#_Toc132920584)

[3 АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ 7](#_Toc132920585)

[4 ФОРМЫ ПРЕДСТАВЛЕНИЯ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ 9](#_Toc132920586)

[5 СЕМАНТИКА И СИНТАКСИС ЯЗЫКА 11](#_Toc132920587)

[6 СТРУКТУРА ПРОГРАММЫ 12](#_Toc132920588)

[7 КОНФИГУРАЦИЯ ТЕХНИЧЕСКИХ СРЕДСТВ 13](#_Toc132920589)

# **1 ВВЕДЕНИЕ**

## **Полное наименование программы**

Многопользовательская система записи долгов.

## **Краткое наименование программы**

«Долговая яма» (далее – программа).

## **Назначение и область применения**

Программа используется пользователем для ведения истории своих выданных и взятых долгов, для синхронизации долговых обязательств между участниками операции.

## **Требования к функционалу**

Программа должна предоставляет пользователю следующие функции:

1. регистрация пользователя;
2. авторизация пользователя;
3. создание долга;
4. погашение долга;
5. удаление долга;
6. редактирование долга;
7. добавление друга;
8. удаление друга;
9. просмотр уведомлений.

# **2 СТРУКТУРА ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ**

На вход программа получает следующие данные:

1. при регистрации пользователя на сервер отправляются: имя, фамилия, отчество, номер телефона пользователя. На клиент приложения отправляется временный код подтверждения авторизации в программе, либо ошибка регистрации;
2. при авторизации пользователя на сервер отправляется: код из СМС сообщения. На клиент возвращается временный код подтверждения пользователя, либо ошибка авторизации;
3. при добавлении, редактировании долга на устройстве сохраняются следующие данные:

- обязательные: сумма долга, статус долга (выдан, взят),

- необязательные: комментарий, второй участник операции.

На сервер отправляется название долга и уникальные идентификационные номера обоих участников операции, сумма задолженности и ее статус;

4) при погашении долга на устройстве сохраняется: сумма погашения долга. На сервер отправляется: название операции, уникальные идентификационные номера участников операции, сумма изменения задолженности;

5) при добавлении друга происходит два сценария:

- добавление друга по его уникальному идентификационному номеру. В этом случае на сервер отправляются уникальный идентификационный номер пользователя и введенный номер. На клиент возвращается подтверждение существования пользователя, добавляемого в друзья, либо ошибка. На устройстве сохраняется введенный номер,

- добавление по номеру телефона. На сервер отправляется номер телефона, на клиент возвращается подтверждение, либо уведомление об ошибке;

6) при удалении друга на сервер отправляются следующие данные: уникальные номера пользователя и друга;

7) при просмотре уведомлений на сервер отправляется запрос на обновление уведомлений на устройстве и уникальный номер пользователя. Сервер присылает данные, либо сообщение об отсутствии новых уведомлений.

# **3 АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ**

Программа запускается на мобильном устройстве.

При нажатии на кнопку «Авторизация» пользователь авторизуется в приложении при помощи кода из СМС-сообщения, отправленного по номеру телефона пользователя. На клиент возвращается временный код подтверждения пользователя, либо ошибка авторизации. Пользователь попадает на «домашнюю страницу».

При нажатии на кнопку «Регистрация» пользователь переходит на экран регистрации. Необходимо заполнить поля имя, фамилия, отчество, номер телефона. Если поля, за исключением отчества, не были заполнены, то пользователю высветится уведомление о необходимости их заполнения, и что пользователь не зарегистрирован. При регистрации пользователя на сервер отправляются: имя, фамилия, отчество (если оно имеется), номер телефона пользователя. На клиент приложения отправляется временный код подтверждения авторизации в программе, либо ошибка регистрации. Пользователь попадает на «домашнюю страницу».

На «Домашней странице» имеются кнопки «Долги», «Добавить долг», «Список друзей», «Уведомления».

При нажатии на кнопку «Добавить долг» пользователь переходит на экран добавления долга. Показываются поля для заполнения: дата выдачи долга, предварительная дата возмещения долга, сумма долга, статус долга, комментарий, второй участник операции. Обязательной информацией является только сумма долга. Если она будут не заполнена, высветится уведомление о необходимости заполнить это поле. Полученная информация сохраняется на телефоне пользователя. Если указан второй участник операции, то на сервер отправляется название долга и уникальные идентификационные номера обоих участников операции, сумма задолженности и ее статус.

При нажатии на кнопку «Удалить долг» выбранный пользователем долг удаляется. Если добавлен второй участник транзакции, то на сервер отправляется уникальный идентификатор транзакции и команда об удалении.

При нажатии на кнопку «Редактировать долг» пользователь переходит на экран выбранного долга, где он может изменять поля: дата выдачи долга, предварительная дата возмещения долга, сумма долга, статус долга, комментарий, второй участник операции. Если указан второй участник операции, то после редактирования долга на сервер отправляется информация об изменении, идентификатор транзакции.

При нажатии на кнопку «Добавление друга» пользователь переходит на экран добавления друга. Для добавления друга доступны следующие функции: «Добавление из списка контактов», «Добавление по номеру телефона», «Добавление по уникальному идентификатору пользователя». Если второй пользователь подтвердит запрос на добавление в друзья, то списки обоих пользователей пополняются. На сервер отправляются уникальный идентификационный номер пользователя и введенный номер. На клиент возвращается подтверждение существования пользователя, добавляемого в друзья, либо ошибка. На устройстве сохраняется введенный номер.

При нажатии на кнопку «Удаление друга» происходит удаление друга из списка друзей пользователя. На сервер отправляются идентификаторы обоих пользователей и команда об удалении их связи.

При нажатии на кнопку «Уведомления» пользователю показываются все недавние уведомления, такие как: заявка на добавление в список друзей, подтверждение долга. При просмотре уведомлений на сервер отправляется запрос на обновление уведомлений на устройстве и уникальный номер пользователя. Сервер присылает данные, либо сообщение об отсутствии новых уведомлений.

# **4 ФОРМЫ ПРЕДСТАВЛЕНИЯ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ**

На вход программа получает следующие данные:

1. при регистрации пользователя на сервер отправляются: имя, фамилия, отчество, номер телефона пользователя. На клиент приложения отправляется временный код подтверждения авторизации в программе, либо ошибка регистрации;
2. при авторизации пользователя на сервер отправляется: код из СМС сообщения. На клиент возвращается временный код подтверждения пользователя, либо ошибка авторизации;
3. при добавлении, редактировании долга на устройстве сохраняются следующие данные:

- обязательные: сумма долга, статус долга (выдан, взят),

- необязательные: комментарий, второй участник операции.

На сервер отправляется название долга и уникальные идентификационные номера обоих участников операции, сумма задолженности и ее статус;

4) при погашении долга на устройстве сохраняется: сумма погашения долга. На сервер отправляется: название операции, уникальные идентификационные номера участников операции, сумма изменения задолженности;

5) при добавлении друга происходит два сценария:

- добавление друга по его уникальному идентификационному номеру. В этом случае на сервер отправляются уникальный идентификационный номер пользователя и введенный номер. На клиент возвращается подтверждение существования пользователя, добавляемого в друзья, либо ошибка. На устройстве сохраняется введенный номер,

- добавление по номеру телефона. На сервер отправляется номер телефона, на клиент возвращается подтверждение, либо уведомление об ошибке;

6) при удалении друга на сервер отправляются следующие данные: уникальные номера пользователя и друга;

7) при просмотре уведомлений на сервер отправляется запрос на обновление уведомлений на устройстве и уникальный номер пользователя. Сервер присылает данные, либо сообщение об отсутствии новых уведомлений.

# **5 СЕМАНТИКА И СИНТАКСИС ЯЗЫКА**

Для реализации визуальной, пользовательской части программы используется кроссплатформенный фреймворк Flutter на языке программирования Dart.

Для реализации серверной части программы используется облачная платформа Firebase от Google, которая предоставляет серверные функции и базу данных в реальном времени.

Firebase предоставляет широкий набор элементов для реализации серверной части приложения, таких как Firebase Cloud Functions, которые используются для обработки запросов от клиентов. Firebase Realtime Database позволяет хранить и синхронизировать данные между клиентами и сервером. Firebase Authentication позволяет авторизовывать пользователей с помощью номера телефона.

# **6 СТРУКТУРА ПРОГРАММЫ**

Программа состоит из фронтенд-части (визуальной, пользовательской) и бэкенд-части (серверной).

Фронтенд-часть программы обеспечивает взаимодействие пользователя и бэкенд-части программы.

Бэкенд-часть программы обеспечивает взаимодействие фронтенд-части программы и СУБД.

Для реализации базы данных программы используется объектно-реляционная СУБД PostgreSQL.

# **7 КОНФИГУРАЦИЯ ТЕХНИЧЕСКИХ СРЕДСТВ**

Операционная система: мобильное приложение может работать на операционных системах iOS или Android, поэтому технические устройства должны быть совместимы с выбранной операционной системой. Минимальная поддерживаемая версия Android должна быть не менее 6.0, а для iOS - не менее 10.

Процессор: устройство должно иметь процессор на базе архитектуры ARMv7 и новее для обеспечения совместимости с операционной системой и приложением.

Оперативная память: мобильное приложение должно иметь достаточно оперативной памяти для быстрой загрузки и обработки данных, поэтому устройство должно иметь не менее 2 ГБ оперативной памяти.

Хранилище: мобильное приложение должно иметь достаточно места для хранения данных, в том числе информации о долгах, поэтому устройство должно иметь не менее 8 ГБ встроенной памяти, а свободное место на устройстве не должно быть менее 120 МБ.

Дисплей: мобильное приложение должно отображаться на дисплее высокого качества с разрешением, соответствующим требованиям приложения, поэтому устройство должно иметь диагональ дисплея не менее 4,7 дюймов и разрешение HD (1280 x 720 пикселей).

Связь: мобильное приложение может использовать мобильные данные или Wi-Fi для связи с Интернетом, поэтому устройство должно иметь поддержку 3G и Wi-Fi с поддержкой стандарта 802.11n.