МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Факультет «Информационные технологии»

Кафедра «Программное обеспечение»

Отчет по лабораторной работе №2

по дисциплине «Проектирование и конструирование ПО»

Выполнили:

студенты группы Б21-191-2 Шитов В.С.

Ценев И.С.

Принял: Еланцев М.О.

Ижевск 2024

1. Прототипы экранных форм
   1. Экран авторизации

Описание: стартовый экран приложения, который появляется при первом запуске. На нем размещены поля для ввода логина и пароля, а так же кнопка регистрации.



Рис.1. Экран авторизации для мобильного приложения



Рис.2. Экран авторизации для десктопного приложения

* 1. Экран регистрации

Описание: экран с полями для ввода никнейма, пароля и электронной почты. Также предусмотрена возможность загрузить аватар для профиля и кнопка, позволяющая вернуться на экран авторизации.

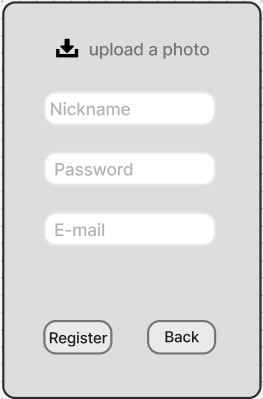


Рис.3. Экран регистрации для мобильного приложения



Рис.4. Экран регистрации для десктопного приложения

* 1. Главная страница в мобильном приложении

Описание: основной экран приложения, включающий меню, список недавних собеседников, сообщества, в которых пользователь состоит, так же есть кнопка создания своего сообщества для общения, текстовый чат с выбранным пользователем, кнопку для совершения личного звонка, а также поле для ввода текста и отправки файлов.

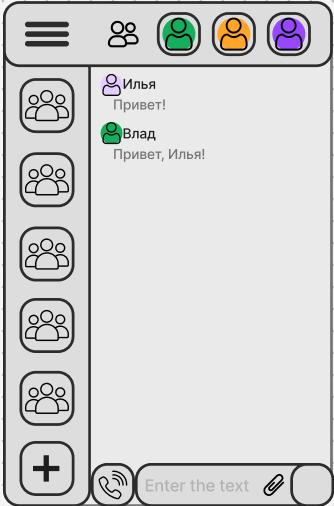


Рис.5. Главная страница в мобильном приложении

* 1. Страница меню в мобильном приложении

Описание: экран, где пользователь может получить доступ к настройкам приложения, добавить друзей по никнейму и просмотреть уведомления.

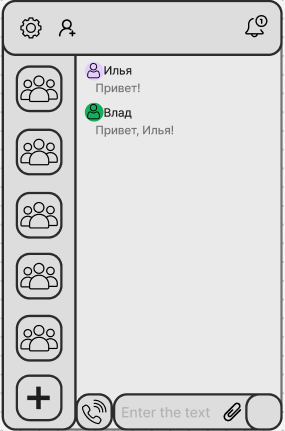


Рис.6. Страница меню в мобильном приложении

* 1. Главная страница десктопного приложения

Описание: основной экран, включающий кнопку настроек, список недавних собеседников, возможность добавления друзей по никнейму, кнопку для просмотра уведомлений, сообщества, в которых состоит пользователь, так же есть возможность создания своего сообщества для общения, текстовый чат с выбранным собеседником, кнопку для совершения личного звонка, поле для ввода текста и отправки файлов. Также отображаются аватар профиля, никнейм, информация о сервере и его отклик.

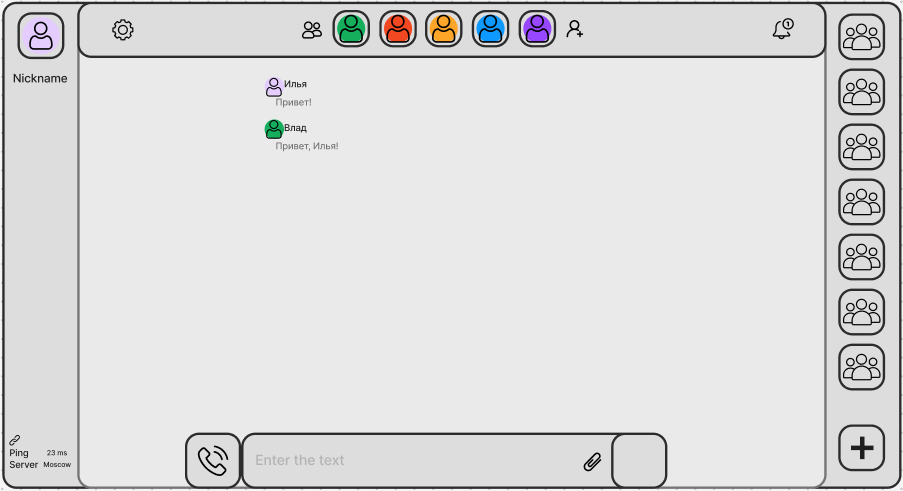


Рис.7. Главная страница в десктопном приложении.

* 1. Интерфейс для личных звонков

Описание: экран, где отображается никнейм собеседника и доступен ряд функций: завершение звонка, отключение звука собеседника, демонстрация экрана, включение веб-камеры, отключение звука устройства и выключение микрофона. При нажатии кнопки "Назад" пользователь возвращается на главную страницу.

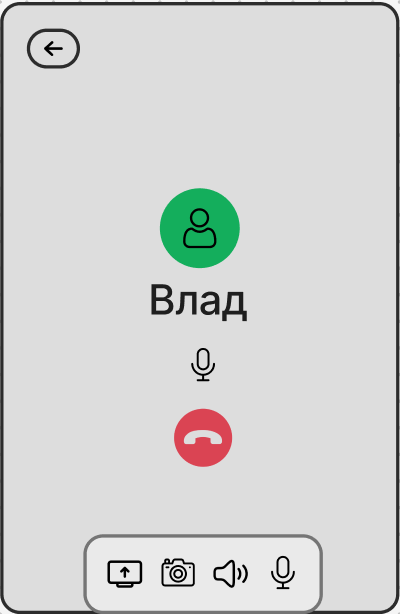


Рис.8. Интерфейс для личных звонков в мобильном приложении

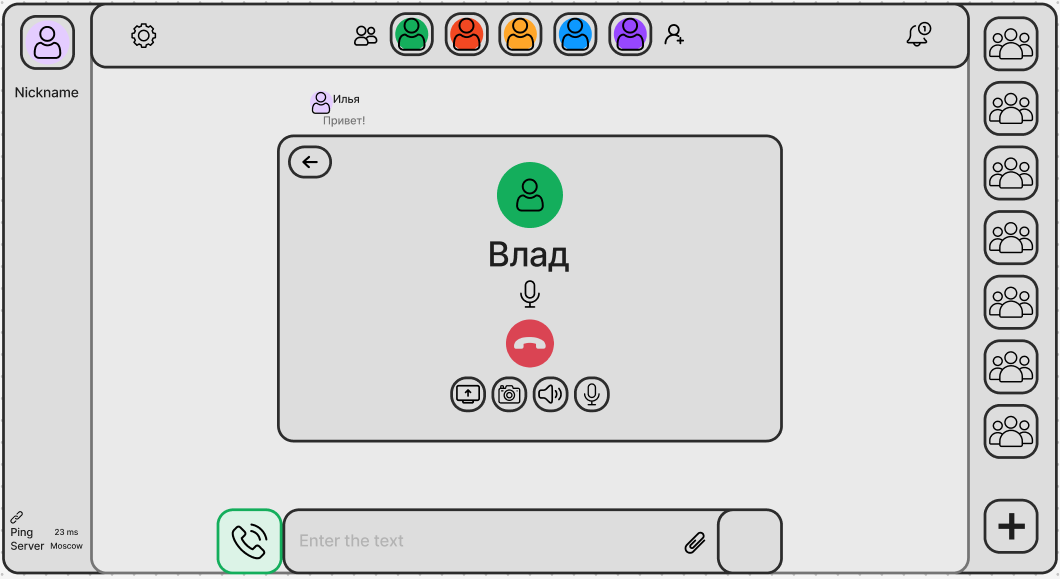


Рис.9. Интерфейс для личных звонков в десктопном приложении

* 1. Интерфейс для голосового сообщества в мобильном приложении  
     Описание: Экран с текстовыми и голосовыми каналами сообщества, где пользователь может выбрать нужный текстовый или голосовой канал из выпадающего списка. Также доступна возможность отключить микрофон.

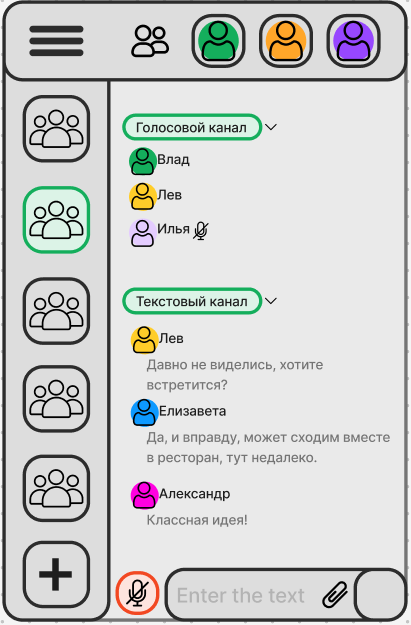


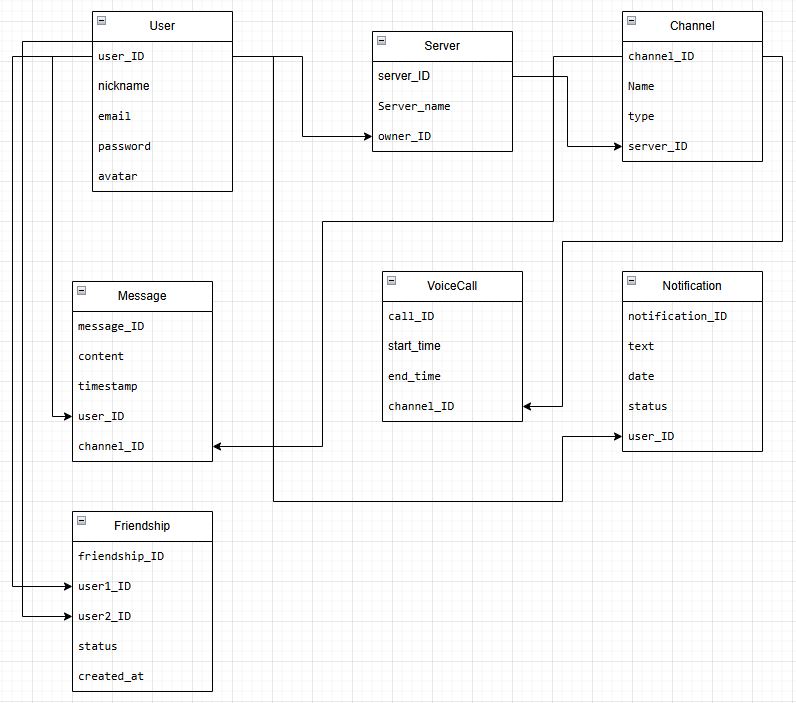
Рис.10. Интерфейс для голосового сообщества в мобильном приложении

1.8 Интерфейс для голосового сообщества в десктопном приложении  
Описание: Экран, отображающий текстовые и голосовые каналы сообщества. Пользователь может отключить микрофон, запустить демонстрацию экрана, включить веб-камеру или отключить звук.



Рис.11. Интерфейс для голосового сообщества в десктопном приложении

1. Диаграмма ER-сущностей



1. Разработка API системы.

1. getUserProfile

Описание действий: Получение профиля пользователя из базы данных.

Входная информация:

userId (int) — идентификатор пользователя

Выходная информация:

Объект класса User — данные о пользователе

2. addUserProfile

Описание действий: Добавление нового пользователя в базу данных.

Входная информация:

Объект класса User — данные пользователя

Выходная информация:

status (bool) — результат операции (успех/ошибка)

3. updateUserProfile

Описание действий: Обновление данных профиля пользователя.

Входная информация:

Объект класса User — обновленные данные пользователя

Выходная информация:

status (bool) — результат операции

4. deleteUserProfile

Описание действий: Удаление профиля пользователя из базы данных.

Входная информация:

userId (int) — идентификатор пользователя

Выходная информация:

status (bool) — результат операции

5. getServersByUser

Описание действий: Получение всех серверов, принадлежащих пользователю.

Входная информация:

userId (int) — идентификатор пользователя

Выходная информация:

Массив объектов класса Server — список серверов пользователя

6. addChannel

Описание действий: Добавление нового канала на сервер.

Входная информация:

Объект класса Channel — данные нового канала

Выходная информация:

status (bool) — результат операции

7. getMessages

Описание действий: Получение сообщений из указанного канала.

Входная информация:

channelId (int) — идентификатор канала

Выходная информация:

Массив объектов класса Message — список сообщений

8. sendMessage

Описание действий: Отправка сообщения в указанный канал.

Входная информация:

Объект класса Message — данные сообщения

Выходная информация:

status (bool) — результат операции

9. getNotifications

Описание действий: Получение всех уведомлений для конкретного пользователя.

Входная информация:

userId (int) — идентификатор пользователя

Выходная информация:

Массив объектов класса Notification — список уведомлений

10. addFriendship

Описание действий: Добавление нового пользователя в список друзей.

Входная информация:

user1Id (int) — идентификатор первого пользователя

user2Id (int) — идентификатор второго пользователя

Выходная информация:

status (bool) — результат операции

11. getFriends

Описание действий: Получение списка друзей пользователя.

Входная информация:

userId (int) — идентификатор пользователя

Выходная информация:

Массив объектов класса User — список друзей

12. startVoiceCall

Описание действий: Создание голосового вызова в голосовом канале.

Входная информация:

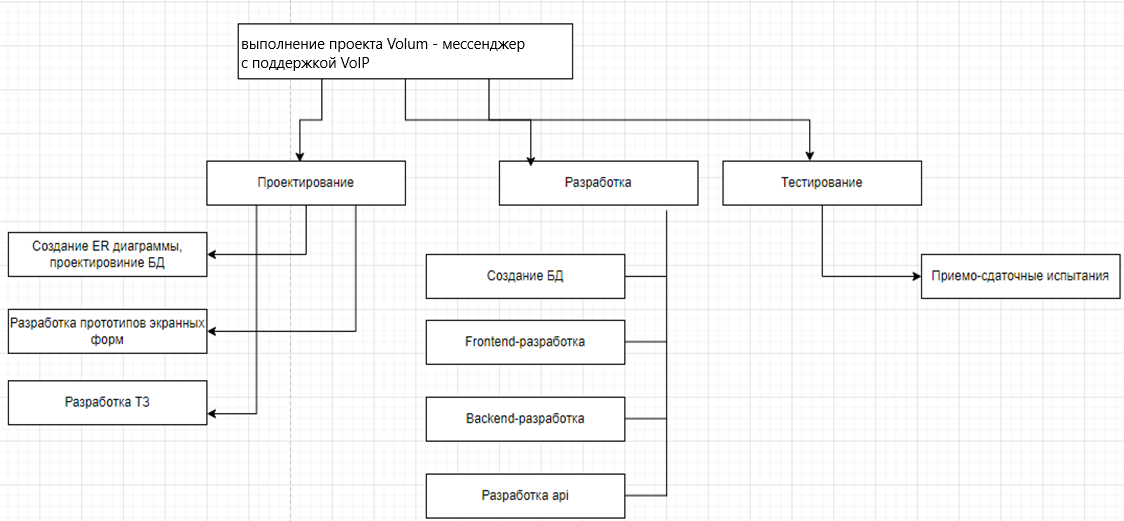
channelId (int) — идентификатор голосового канала

userId (int) — идентификатор инициатора вызова

Выходная информация:

Объект класса VoiceCall — данные о вызове

1. Иерархическая структура работа (ИСР)



1. Расчет времени по методу PERT.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | N | O | M | P | E | СКО |
| Создание ER-диаграммы и проектирование БД | 1 | 10 | 18 | 21 | 17.2 | 1,83 |
| Разработка прототипов экранных форм | 7 | 2 | 3 | 4 | 3 | 0.33 |
| Разработка технического задания (ТЗ) | 1 | 7 | 8 | 9 | 8 | 0.33 |
| Создание базы данных | 1 | 10 | 13 | 15 | 12.8 | 0.83 |
| Frontend-разработка | 7 | 7 | 8 | 9 | 8 | 0.33 |
| Разработка API | 10 | 5 | 7 | 8 | 6.83 | 0.5 |
| Приемо-сдаточные испытания | 1 | 4 | 5 | 6 | 5 | 0.33 |

Формула:

Общие трудозатраты:

E = 17.2+21+8+12.8+56+68.3+5=188.3 часов

Среднее квадратическое отклонение:

Итоговое

Итак, общие трудозатраты проекта составляют 195 часов по методу PERT.

1. Диаграмма Ганта