**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра технологий программирования

Миронова Анна Викторовна

Отчет по лабораторной работе No4 курса  
«Технологии программирования» студентки 2 курса 7 группы

Преподаватель  
Карпович Наталья Александровна

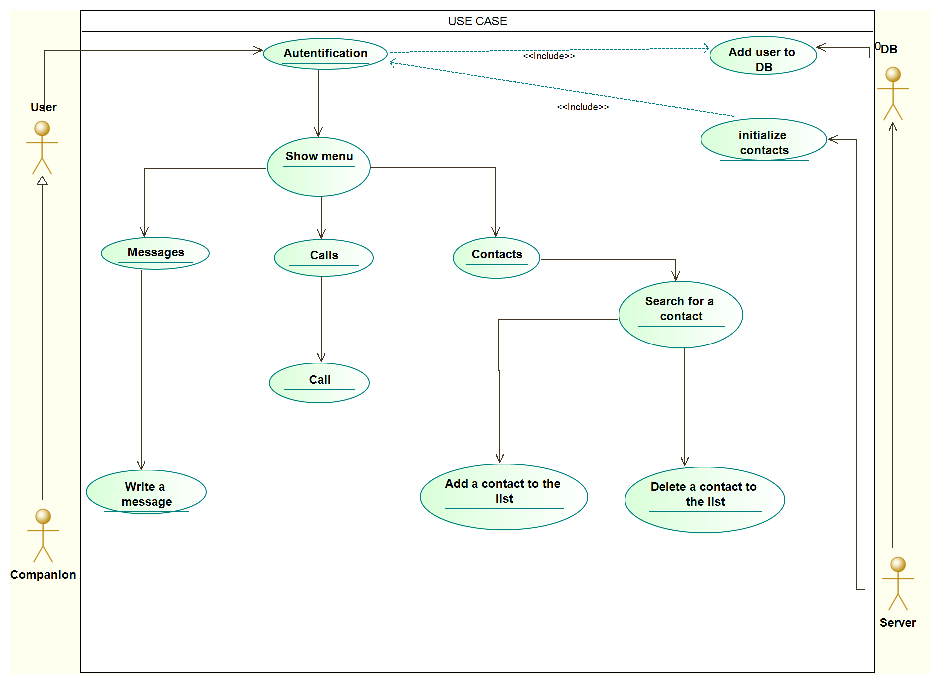
**Минск 2023**

**Описание предметной области** (из варианта задания)

В приложении пользователи обмениваются сообщениями.

* Для того чтобы общаться пользователь должен зарегистрироваться, при этом фиксируются его  
  ФИО, телефон и email, логин и пароль.
* Пользователь должен сформировать свой список для  
  общения : либо добавить по логину или номеру телефона собеседников, либо собеседники сами  
  добавляются после отправки пользователю сообщения.
* Сообщения можно редактировать и  
  удалять, подключать файлы.
* Можно создавать групповой чат. В приложении можно осуществлять  
  звонки на номер собеседника.

Необходимо осуществлять следующую обработку данных:  
- список собеседников;  
- редактирование настроек приложения;  
- список звонков.



Концептуальная модель системы с обновленными (исправленными) диаграммой (и) вариантов использования и сценариев.

**Глоссарий**

|  |  |
| --- | --- |
| Пользователь(User) | Пользователь, регистрирующийся в сети для мгновенного обмена сообщениями и обладающий возможностью добавлять других пользователей в список контактов и обмениваться с ними данными. |
| Собеседник  (Companion) | Пользователь, регистрирующийся в сети для мгновенного обмена сообщениями и обладающий возможностью добавлять других пользователей в список контактов и обмениваться с ними данными. |
| Сообщения  (Messages) | Наименьший элемент языка, имеющий идею или смысл, пригодный для общения. Функционал сервиса, который дает возможность обмениваться сообщениями. |
| Звонки(Calls) | Функционал сервиса, который обеспечивает связь с другими пользователями |
| Контакты( Contacts) | Список всех собеседников пользователя. |
| Меню (Menu) | Интерфейс для интуитивного использования функционала. |
| Сервер ПО  (Server) | Сервер ПО принимающее запросы от пользователя, при регистрации пользователя делает запись в базу данных. |
| База данных(DB) | База данных Содержит информацию о всех пользователях и ихсвязях между собой. |

**Спецификация прецедента «Аутентификация»:**

|  |
| --- |
| Прецедент: Аутентификация |
| ID: 1 |
| Краткое описание: Пользователь входит в сеть под своим логином |
| Главные актеры: Пользователь |
| Предусловия: Прецедент начинается, когда пользователь выбрал опцию «Аутентификация» |
| Основной поток: 1. Пользователь вводит логин и пароль в специальное поле 2. Сервер проверяет соответствие пароля и логина в базе данных |
| Постусловия: 1. Сервер инициализирует контакты для пользователя и возвращает ему данные |

**Спецификация прецедента «Регистрация»:**

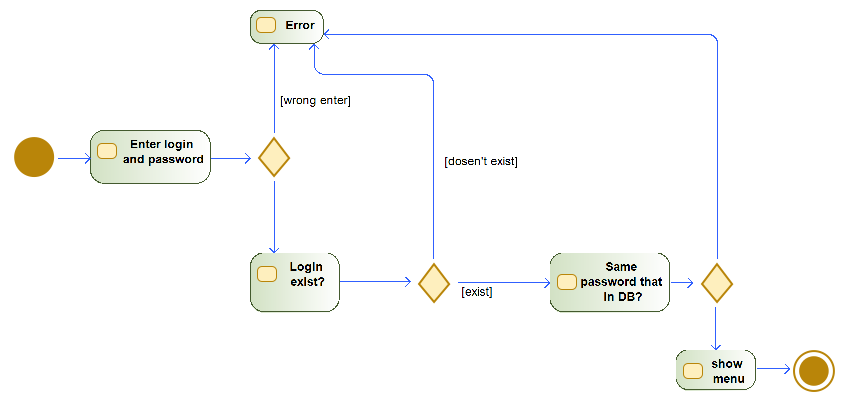
|  |
| --- |
| Прецедент: Регистрация |
| ID: 2 |
| Краткое описание: Регистрация нового пользователя |
| Главные актеры: Пользователь |
| Предусловия: Прецедент начинается, когда пользователь выбрал опцию «Регистрация» |
| Основной поток: 1. Пользователь вводит логин, адрес электронной почты и пароль для своего аккаунта в специальное поле 2. Пользователь подтверждает регистрацию, ответив на письмо, отправленное ему на электронную почту |
| Постусловия: 1. В базе данных зарегистрирован новый пользователь |

**Спецификация прецедента «Занести пользователя в базу данных»:**

|  |
| --- |
| Прецедент: Занести пользователя в базу данных |
| ID: 3 |
| Краткое описание: Информация о зарегистрированном пользователе заносится в базу данных |
| Главные актеры: Сервер |
| Второстепенные актеры: База данных |
| Предусловия: Прецедент начинается, когда пользователь выбрал опцию «Регистрация» |
| Основной поток: 1. Сервер вызывает процедуру добавления в базу данных |
| Постусловия: 1. В базе данных зарегистрирован новый пользователь |

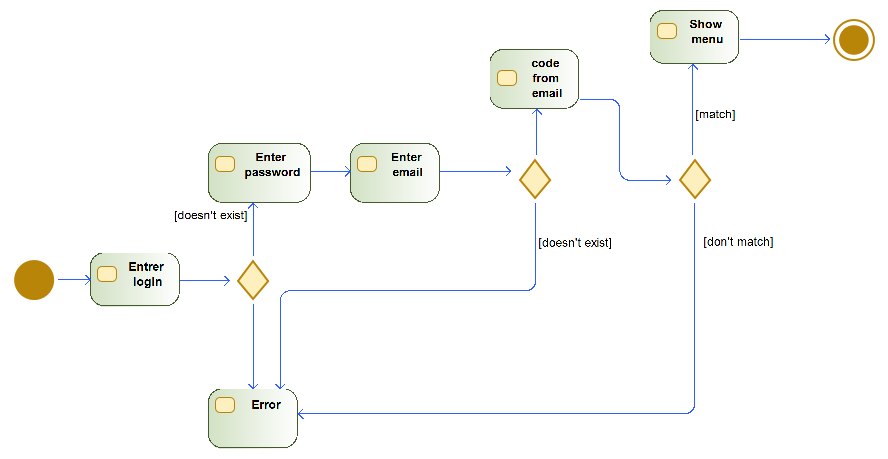
**Диаграммы деятельности/активности.**

**«Аутентификация»:**

****

|  |
| --- |
| Аутентификация |
| Краткое описание: Пользователь входит в сеть под своим логином |
| Главные актеры: Пользователь |
| Предусловия: Прецедент начинается, когда пользователь выбрал опцию «Аутентификация» |
| Основной поток: 1. Пользователь вводит логин и пароль в специальное поле 2. Сервер проверяет соответствие пароля и логина в базе данных |
| Постусловия: 1. Сервер инициализирует контакты для пользователя и возвращает ему данные |

**«Регистрация»:**

****

|  |
| --- |
| Регистрация |
| Краткое описание: Регистрация нового пользователя |
| Главные актеры: Пользователь |
| Предусловия: Прецедент начинается, когда пользователь выбрал опцию «Регистрация» |
| Основной поток: 1. Пользователь вводит логин, адрес электронной почты и пароль для своего аккаунта в специальное поле 2. Пользователь подтверждает регистрацию, ответив на письмо, отправленное ему на электронную почту |
| Постусловия: 1. В базе данных зарегистрирован новый пользователь |

**Диаграммы классов.**

Классы в данной предметной области могут быть представлены следующим образом:

1. Класс Пользователь (Class User):

* Имя (string)
* Фамилия (string)
* Телефон (string)
* Email (string)
* Логин (string)
* Пароль (string)
* Список контактов (Contact[])

1. Класс Контакт (Contact):

* Логин (string)
* Телефон (string)

1. Класс Сообщение(Message):

* Текст (string)
* Время отправки (datetime)
* Автор (Пользователь)
* Получатель (Пользователь/Группа)

1. Класс Группа(Group):

* Название (string)
* Участники (Пользователь[])

1. Класс Файл(File):

* Имя файла (string)
* Размер файла (int)
* Содержимое (byte[])

1. Класс Звонок(Call):

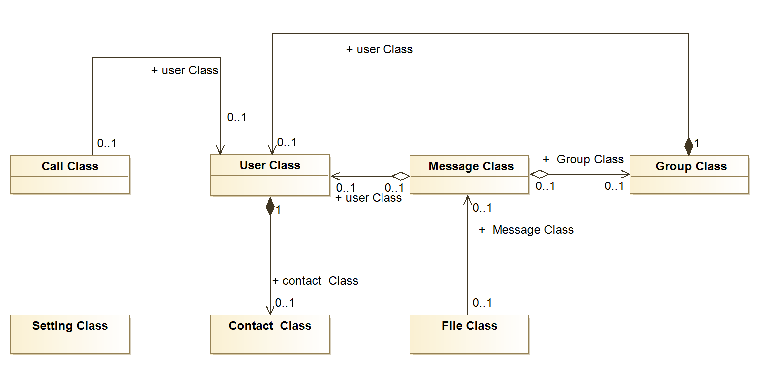
* Время начала звонка (datetime)
* Длительность звонка (time)
* Отправитель (Пользователь)
* Получатель (Пользователь)

1. Класс Настройки(Settings):

* Язык (string)
* Размер шрифта (int)
* Тема оформления (string)

| **Класс** | **Атрибуты** | **Методы** |
| --- | --- | --- |
| Пользователь | Имя, Фамилия, Телефон, Email, Логин, | getContacts(): Contact[] |
|  | Пароль, Список контактов | editProfile(name, phone, email, password): void |
| Контакт | Логин, Телефон |  |
| Сообщение | Текст, Время отправки, Автор, Получатель | edit(text): void, delete(): void |
| Группа | Название, Участники | addMember(user): void, removeMember(user): void |
| Файл | Имя файла, Размер файла, Содержимое |  |
| Звонок | Время начала звонка, Длительность звонка, Отправитель, Получатель |  |
| Настройки | Язык, Размер шрифта, Тема оформления | editLanguage(language): void, editFontSize(fontSize): void, editTheme(theme): void |

**Общая диаграмма системы.**



Связи между классами:

1. Класс User имеет отношение композиции с классом Contact, поскольку каждый пользователь имеет список контактов.
2. Класс Message имеет отношение агрегации с классами User и Group, поскольку каждое сообщение имеет автора и получателя, которые могут быть как пользователем, так и группой.
3. Класс Group имеет отношение композиции с классом User, поскольку каждая группа имеет список участников, которые являются пользователями.
4. Класс Call имеет отношение ассоциации с классом User, поскольку каждый звонок включает двух пользователей - вызывающего и принимающего.
5. Между классом Settings и другими классами нет прямых связей, поскольку он хранит настройки приложения и не связан напрямую с функциональностью обмена сообщениями и звонков.
6. Класс File не имеет прямых связей с другими классами, но он может быть связан с сообщением, так как файлы могут быть прикреплены к сообщениям.

