*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка программной системы   
информационной системы «Мессенджер»*

*Выполнил: студент гр. ПРИ-120*

*Бочков М. А.*

*Принял: доц. кафедры ИСПИ*

*Вершинин В.В.*

*Владимир, 2022*

**<утвержденный Лист задания нА курсовой проект>**

**Аннотация**

<<заглушка>>

**Содержание**

ВВЕДЕНИЕ 3

1 Описание предметной области 4

1.1. Пользователи разрабатываемой подсистемы 4

1.2. Начальная оценка и выделение сущностей 4

1.3. Словарь предметной области 4

2. проектирование системы 12

заключение 13

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14

ПРИЛОЖЕНИЕ А Исходный код программной системы 15

ПРИЛОЖЕНИЕ Б Структура таблиц БД 16

# ВВЕДЕНИЕ

<<заглушка>>

# Описание предметной области

Мессенджер – программа для обмена информацией в реальном времени.

Каждый мессенджер должен иметь как минимум 2-ух пользователей. Также должен быть чат для обмена информацией между пользователями и должен включать в себя оповещение о получении сообщения. Некоторые мессенджеры поддерживают функцию добавления в друзья.

Также некоторые мессенджеры включают в себя создание групповых чатов.

* 1. Пользователи разрабатываемой подсистемы

*Пользователь –* может общаться с другими пользователями, добавлять в друзья и создавать групповые чаты.

* 1. Начальная оценка и выделение сущностей

Для предметной области «Мессенджер» были выделены следующие сущности: аккаунт, профиль, друзья, личные сообщения, групповой чат, групповые сообщения.

* 1. Словарь предметной области

*Аккаунт (Account) —* создание аккаунтов.

Атрибуты: уникальный номер, почта пользователя, пароль пользователя

*Пользователь (User) —* Профиль пользователя, который отображает информацию о нём.

Атрибуты: уникальный номер, номер аккаунта, имя и фамилия пользователя, дополнительная информация.

*Друг (friend)* — состояние пользователя по отношению к другому пользователю.

Атрибуты: уникальный номер, номер первого пользователя, номер второго пользователя, статус.

*Групповой чат* – чат, поддерживающий общение между несколькими пользователями.

Атрибуты: уникальный номер, номер группового чата, номер пользователя

*Групповое сообщение (GroupMessage)* — сообщение в групповом чате.

Атрибуты: уникальный номер, номер пользователя, номер группового чата, текст сообщения, дата.

*Личное сообщение (PrivateMessage) –* сообщение в личной переписке.

Атрибуты: уникальный номер, номер первого пользователя, номер второго пользователя, текст сообщения, дата.

Описание прецедентов

**Название: “**Регистрация”

**Предусловие:** -

**Действующее лицо:** Пользователь

**Основной поток:** Регистрация пройдена

Данные о пользователя сохранятся в базе данных и будут использованы в дальнейшем.

**Альтернативный поток:** Ошибка при регистрации

На шаге 2 пользователь не смог пройти регистрацию, по причине уже существующего пользователя с таким никнеймом.

**Постусловие:** Пользователь сможет пройти авторизацию

**Название: “**Авторизация”

**Предусловие:** Пользователь должен пройти регистрацию

**Действующее лицо:** Пользователь

**Основной поток:** Пользователь авторизируется в системе.

Данные о пользователя сверяются с данными из базы данных, и в случае нахождения соответствия, пользователь войдет в систему.

**Альтернативный поток:** Ошибка при авторизации

На шаге 2 пользователь не смог пройти авторизацию, по ряду возможных причин:

1. Не прошел регистрацию
2. Ошибка при вводе данных.

**Постусловие:** Пользователь войдет в систему.

**Название: “**Поиск людей”

**Предусловие:** Пользователь должен быть авторизирован

**Действующее лицо:** Пользователь

**Основной поток:** Пользователь получает список всех людей

Пользователь получает список всех людей, котором он может отправить сообщение, либо добавить в друзья.

**Альтернативный поток:** Отсутствие людей в системе

На шаге 2 пользователь смог воспользоваться поиском, но при этом список людей был пуст.

**Постусловие:** Пользователь сможет отправить сообщение, либо добавить в друзья

**Название: “**Отправлять сообщения”

**Предусловие:** Пользователь должен быть авторизирован,

**Действующее лицо:** Пользователь

**Основной поток:** Сообщение будет отправлено

Пользователь набирает свое сообщение, которое будет отправлено.

**Альтернативный поток:** Ошибка при отправке

На шаге 2 пользователь не смог отправить сообщение, в связи с техническими неполадками, либо при попытке отправить пустое сообщение. /-

**Постусловие:** Другой пользователь получает сообщение

**Название: “**Получать сообщения”

**Предусловие:** Пользователь должен быть авторизирован, другой пользователь должен отправить текущему сообщение.

**Действующее лицо:** Пользователь

**Основной поток:** будет получено сообщение

Пользователь получает сообщение от другого пользователя

**Альтернативный поток:** Ошибка при получении

На шаге 2 пользователь не смог получить сообщение, в связи с техническими неполадками.

**Постусловие:** Пользователь получает оповещение.

**Название: “**Получать оповещение”

**Предусловие:** Пользователь должен быть авторизирован, другой пользователь должен отправить текущему сообщение/заявку в друзья/создать с текущим пользователем групповой чат.

**Действующее лицо:** Пользователь

**Основной поток:** будет получено оповещение

Пользователь получает оповещение о полученном сообщении/о том, что пришла заявка в друзья / о том, что другой пользователь создал с ним групповой чат.

**Альтернативный поток:** Ошибка при получении

На шаге 2 пользователь не смог получить сообщение, в связи с техническими неполадками.

**Постусловие:** -

**Название: “**Создать групповой чат”

**Предусловие:** пользователь должен быть авторизирован,

**Действующее лицо:** Пользователь

**Основной поток:** будет создан групповой чат

Пользователь приглашает других пользователей в групповой чат

**Альтернативный поток:** Другой пользователь не будет добавлен

На шаге 2 пользователь не смог авторизоваться, в связи с этим ему стоит проверить данные, либо пользователь пытается добавить другого пользователя, который не является для него другом. /-

**Постусловие:** Другие пользователи будут добавлены в групповой чат, и они получат оповещение

**Название: “**Добавить в друзья”

**Предусловие:** пользователь должен быть зарегистрирован и авторизирован в системе в системе.

**Действующее лицо:** Пользователь

**Основной поток:** принять заявку в друзья

В БД с таблицей “Пользователь” будет связана таблица “Друзья”, в которую будут заноситься данные пользователя, которого добавляют в друзья.

**Альтернативный поток:** отклонить заявку в друзья

Пользователь, которого добавляют в друзья, может отклонить приглашение, в этом случаи данные не будут добавляться в БД

**Постусловие:** Пользователь, который добавлял в друзья другого пользователя, получит соответствующее оповещение.

Диаграмма UseCase:



Диаграмма классов:

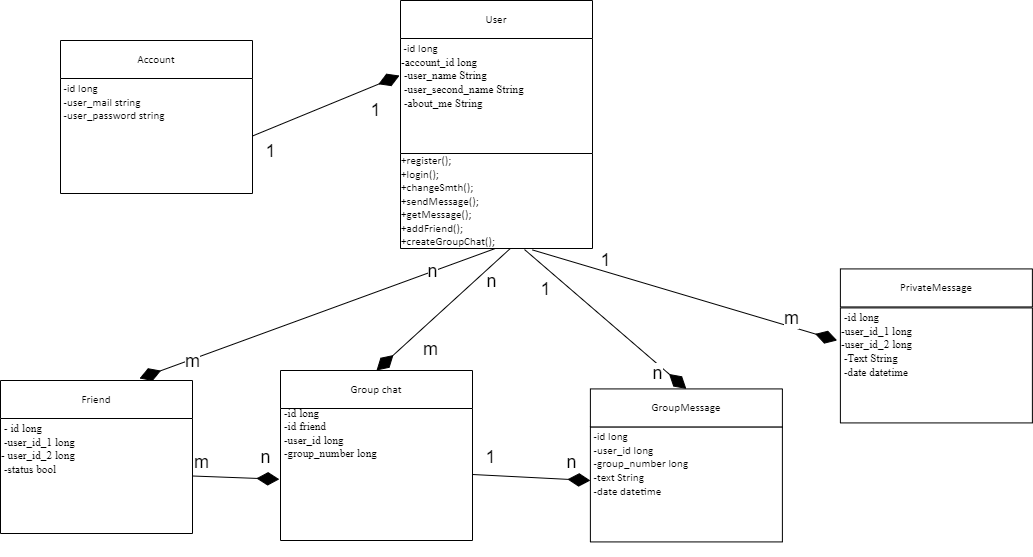


Диаграмма состояния “Друзья”

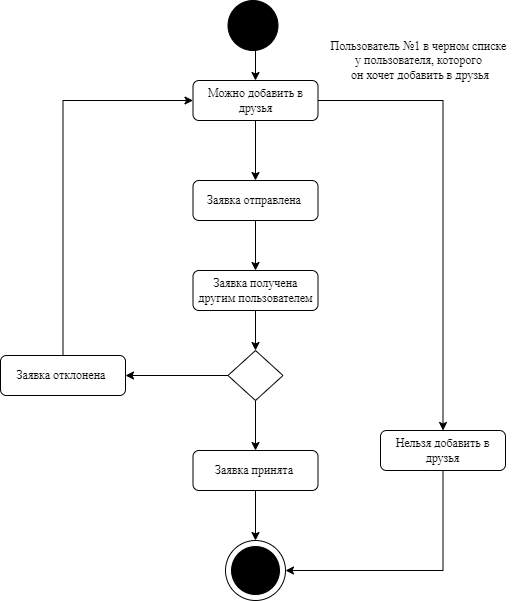
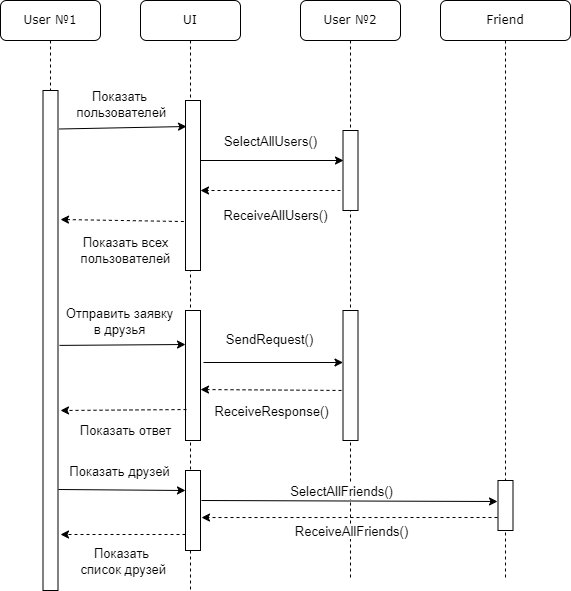


Диаграмма последовательности “Добавление в друзья”



# проектирование системы

Проект находится по следующей ссылке:

https://github.com/Gladius4413/KP-TP

# заключение

Основные выводы по работе и достигнутые результаты. Достаточно трех четырех абзацев текста.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Липаев В.В. Проектирование программных систем. М.: Высш. шк, 1990.

2. Буч Г. Объектно-ориентированное проектирование / Пер. с анг. Конкорд, 1996.

3. Майерс Г. Надежность программного обеспечения. М.: Мир, 1980

4. <название статьи/книги/материала> [Электронный ресурс]: <автор>. – Режим доступа: <полный URL>

ПРИЛОЖЕНИЕ А Исходный код программной системы

package ru.home.lt.cf;

import lrapi.lr;

import org.apache.commons.configuration.ConfigurationException;

import org.apache.commons.configuration.XMLConfiguration;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class UC01CreateApplication extends UseCase{

final static private Logger logger = LoggerFactory.getLogger(UC01CreateApplication.class);

public static Double createILSFPercentage = 82.82;

public static Double createCWFPPercentage = 10.80;

public static Double createLLQ1Percentage = 5.21;

public static Double createCORNPercentage = 0.78;

public static Double createCCLPPercentage = 0.39;

private UC01CreateILSF ilsf;

private UC01CreateCWFP cwfp;

private UC01CreateLLQ1 llq1;

private UC01CreateCORN corn;

private UC01CreateCCLP cclp;

static {

XMLConfiguration config = null;

try {

config = new XMLConfiguration("../load-test.xml");

} catch (ConfigurationException e) {

logger.error(e.getLocalizedMessage());

}

createILSFPercentage = config.getDouble(LoadTestConfiguration.CREATE\_ILSF\_APPLICATION\_PERCENTAGE, 82.82);

createCWFPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CWFP\_APPLICATION\_PERCENTAGE, 10.80);

createLLQ1Percentage = config.getDouble(LoadTestConfiguration.CREATE\_LLQ1\_APPLICATION\_PERCENTAGE, 5.21);

createCORNPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CORN\_APPLICATION\_PERCENTAGE, 0.78);

createCCLPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CCLP\_APPLICATION\_PERCENTAGE, 0.39);

LogProductPercentages();

}

ПРИЛОЖЕНИЕ Б Структура таблиц БД

**карман с диском на котором записаны материалы курсового проектирования:**

1. пояснительная записка

2. проект системы с исходными кодами

3. дополнительные компоненты системы, необходимые для ее развертывания

4. прочие вспомогательные источники

5. презентация по которой будет проходить защита работы

**Папка(файлик) для графического материала к пояснительной записке и сам графический материал в ней**