УДК 004.056.52

Н. С. Андращук
Учреждение образования «Барановичский государственный университет», Барановичи

СЕТЕВОЕ ПРИЛОЖЕНИЕ «ЧАТ-МИКРОБЛОГ»

Введение. В настоящее время в современном мире электронных технологий практически невозможно представить жизнь без мессенджеров. За последние несколько лет скорость передачи данных, и доступности сети Internet достигла очень высокого уровня, дав толчок для бурного роста и популяризации данного типа приложений. Телефонные разговоры отнимают много времени и не всегда бывают уместны, а вот мессенджеры —хорошая альтернатива.

В основе мессенджеров лежит асинхронность, долговременность и реальность диалогов между людьми. Сочетание всех этих свойств и ожиданий делает их основным цифровым средством коммуникации. Это самая удобная среда для связи из когда-либо изобретенных.

Основная часть. Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине.

Клиент — компьютер, осуществляющий запрос к серверу на выполнение каких-либо действий или предоставление какой-либо информации. Сервер — компьютер, обычно более мощный, чем компьютер-клиент. Модель функционирования такой системы заключается в следующем: клиент делает запрос серверу, сервер (серверная часть) получает запрос, выполняет его и отсылает результат клиенту (клиентская часть).

Сервер может обслуживать нескольких клиентов одновременно. Запросы выполняются сервером последовательно. Если одновременно приходит более одного запроса, то запросы устанавливаются в очередь. В данном случае очередь — это список невыполненных клиентских запросов. Иногда запросы могут иметь приоритеты. Приоритет — это уровень важности выполнения запроса. Запросы с более высокими приоритетами должны выполняться раньше [1].

Существует концепции построения системы клиент-сервер:

Слабый клиент — мощный сервер — вся обработка информации осуществляется целиком сервером. Сервер посылает готовый результат, не требующий дополнительной обработки. Клиент только ведет диалог с пользователем: составляет запрос, отсылает запрос, принимает запрос и выводит информацию на экран (на принтер, в файл) [2].

Сильный клиент — часть обработки информации перепоручается клиенту.

Целью данного проекта является создание клиент-серверного приложения для коммуникаций в групповых чатах и ведении собственного блога — Чата-Микроблога.

Данное программное обеспечение позволит создавать и управлять собственными чатам и блогами, вступать в другие групповые чаты, иметь подписку на чей-либо личный блог. Все передаваемые конфиденциальные данные безопасно передаются на сторону сервера и хранятся там в зашифрованном виде. Управление приложением представлено в виде удобного графического интерфейса.

Чат-Микроблог — это все необходимое для комфортной коммуникации с большой группой лиц.

Созданное приложение позволяет общаться и доносить информацию большому количеству человек в удобном формате.

Все классы, используемые в приложении, хранятся в пакетах, структурированных папках, содержащих классы, предназначенные для выполнения одной задача, представлена на рисунке 1.

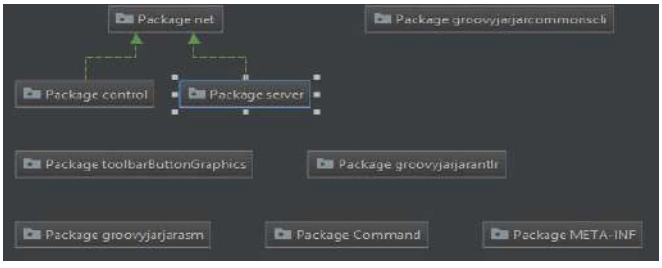

Рисунок 1 — Диаграмма пакетов

Сокет — название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут исполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет — абстрактный объект, представляющий конечную точку соединения. Сокеты используются при подключения нового клиента, сервера, для отправки данных на сервер и т.д. [3].

JSON, что означает JavaScript Object Notation, — это текстовый формат обмена данными, который легко читается человеком и в то же время является компактным (в отличии от того же XML формата). JSON произошел от javascript и очень часто используется в веб-программировании при обмене данными между веб-браузером и сервером.

Json — это небольшая java-библиотека, которая позволяет конвертировать java-объекты в их JSON-представление, равно как и создавать объекты на основании их json-представления [4].

Основным классом библиотеки есть одноименный класс Json. Для того, чтобы создать экземпляр класса нужно воспользоваться одним из двух способов:

Gson gson = new Gson();

Gson gson = new GsonBuilder().create();

Основные методы, которые используются для сериализации и десериализации java-объектов, называются toJson и fromJson .

Шифрование — обратимое преобразование информации в целях сокрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней. Главным образом, шифрование служит задачей соблюдения конфиденциальности передаваемой информации. Важной особенностью любого алгоритма шифрования является использование ключа, который утверждает выбор конкретного преобразования из совокупности возможных для данного алгоритма.

RSA — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Для работы с этим ассиметричным алгоритмом в Java представлена библиотека java.security [5].

Диаграмма Use Case (Вариант использования) описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату. Варианты использования являются описаниями типичных взаимодействий между пользователями системы и самой системой. Они отображают внешний интерфейс системы и указывают форму того, что система должна сделать (именно что, а не как).

Диаграмма Use Case представлена на рисунке 2.

Сервер реализован в виде консольного приложения, что позволяет без проблем запускать его на удаленных серверных машинах, также он позволяет администратору просматривать состояние подключенных пользователей в реальном времени.

Для запуска сервера необходимо выполнить скрипт командной строки «ServerStart». Вид консольного окна отображен на рисунке 3.

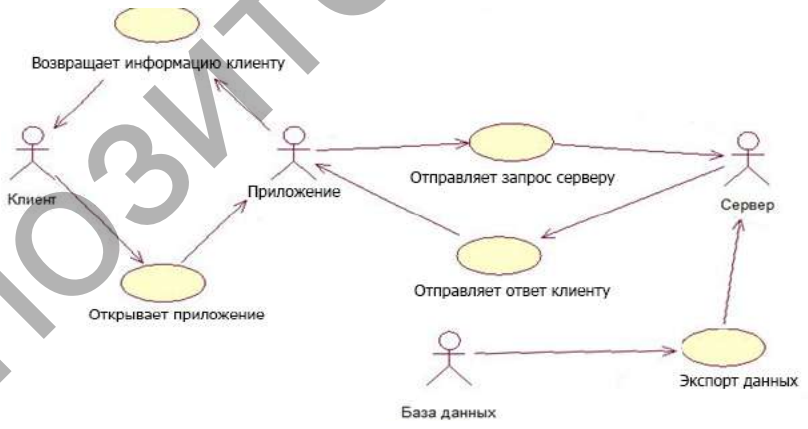

Рисунок 2 — Диаграмма Use Case

```
Server started...
Connection to DataBase Completed!
Network connection set to port 8190
Client /127.0.0.1: 50251 connection.
```

Рисунок 3 — Представление сервера в приложении

Клиент представлен в виде главной формы. Для работы с программой вначале необходимо авторизоваться — ввести логин, пароль и нажать на кнопку «Войти» (рисунок 4).

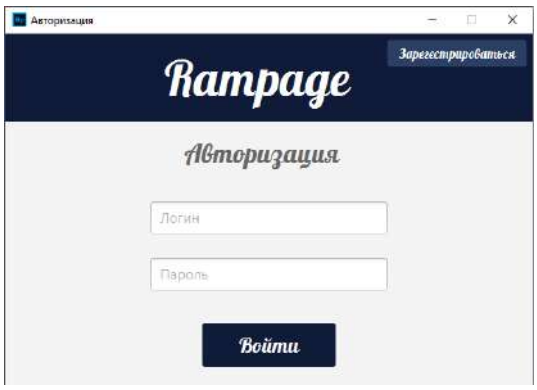

Рисунок 4 — Авторизация

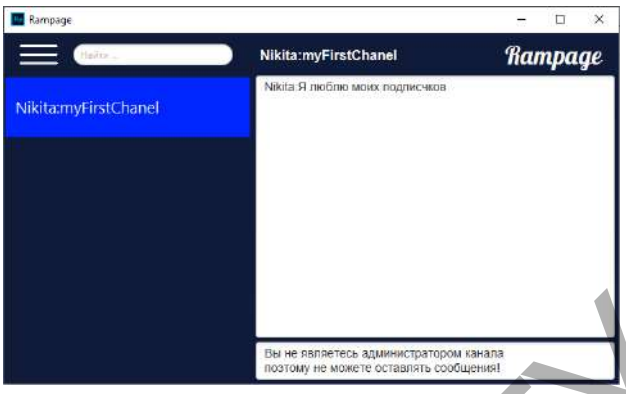

Рисунок 5— Главное окно

После успешного входа в систему перед пользователем открывает главное окно приложения, на котором расположен список чатов пользователя, их поиск, меню, поле ввода и просмотра текста (рисунок 5).

Заключение. Программа Чат-Микроблог была написана на строго типизированном объектно-ориентированном языке программирования Java с использованием всех возможностей данного языка. Было проведено тестирование программы, позволяющее увидеть весь ее функционал, преимущества и недостатки. Данное приложение показало важность таких составляющих разработки как frondend и backend (клиентской и серверной части).

Список цитируемых источников

1. Объектно-ориентированный подход [Электронный ресурс]. — Режим доступа: http://opensourcerules.net/java/gl3_2.html. Дата доступа: 13.12.2018

2. Евсеева, О. Н. Работа с базами данных на языке JAVA. / О. Н. Евсеева, А. Б. Шамшев. — Ульяновск: УлГТУ, 2009. — 170 с.

3. Сокеты — сетевое программирование [Электронный ресурс]. — 2016. — Режим доступа: http://lecturesnet.readthedocs.io/net/low-level/ipc/socket/intro.html. — Дата доступа 22.12.2018.

4. Обзор JSON — сетевое работает с JSON Java [Электронный ресурс]. — 2016. — Режим доступа: http://www.javenue.info/post/gson-json-api — Дата доступа 27.12.2018.

5. Объектно-ориентированное программирование [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование. Дата доступа: 24.12.2018.

УДК 519.1

В. С. Бурмако, Д. С. Кислый

Учреждение образования «Барановичский государственный университет», Барановичи

ШИФРОВАНИЕ КАК МЕТОД ЗАЩИТЫ ИНФОРМАЦИИ

Введение. Необходимость засекречивать важные послания возникла еще в древние времена. Со временем люди находили все более сложные способы делать послания недоступными чужим глазам. Древние рукописи и языки были поняты с помощью техник декодирования и дешифрования. Самый известный пример — Розеттский камень Древнего Египта. Фактически коды и шифры определяли исход многих войн и политических интриг на протяжении всей истории человечества [1].

Актуальность темы очевидна, так как информация в современном обществе — одна из самых ценных вещей в жизни, требующая защиты от проникновения лиц, не имеющих к ней доступа. Что такое шифрование? Шифрование — это способ преобразования пригодного для чтения текста, делающий невозможным его прочтение третьими лицами, причем текст снова становится пригодным для чтения после верификации ключа. Таким образом, суть шифрования заключается в том, что зашифрованная информация не представляет никакой ценности без знания ключа доступа. Криптография — наука о шифрах. Сообщения шифруются и расшифровываются с помощью алгоритмов, созданных комбинацией информатики и математики. С развитием технологий человечество старается находить все более надежные способы защиты информации. Но для того, чтобы понять более сложные методы шифрования, необходимо знать и понимать, как работают шифры, изобретенные еще до появления компьютеров. Хотя растущая производительность компьютеров позволяет